



MicroPatent® PatSearch Fulltext: Record 1 of 1

Search scope: JP (bibliographic data only)

Years: 1836-2006

Patent/Publication No.: ((JP10078918))

Order/Download

Family Lookup

Find Similar

Legal Status

Go to first matching text

JP10078918 A
CHECK POINT PROCESSING
ACCELERATOR AND COMPUTER
SYSTEM TO WHICH THE SAME
DEVICE IS APPLIED
TOSHIBA CORP

Abstract:

PROBLEM TO BE SOLVED: To provide a check point processing accelerator in which a check point/recovery function can be easily realized in a computer on which a standard uni-processor or the like is mounted.

SOLUTION: When data update is operated on a cache memory 20, a before- image obtaining part 32 obtains the update address and data before updated, and stores them in a before-image storing part 35. A cache flash executing part 34 issues a command for requesting the rewriting of the content of a cache block in an updated state to a main memory 51 to a system bus 40 by using all the addresses stored in the before-image storing part 35. On the other hand, a main memory state restoring part 33 issues a command for requesting update writing in the main memory 51 to the system bus 40 related with the data before updated stored in the before-image storing part 35 in the order of the newly stored data.

[no drawing]

Inventor(s):

KANO TAKUYA
SAKAI HIROSHI
HAYASHI HIROO

Application No. 08234321 JP08234321 JP, **Filed** 19960904, **A1 Published** 19980324

Original IPC(1-7): G06F01208
G06F01114 G06F01216

Current IPC-R	invention	version	additional	version
Advanced	G06F01114	20060101		
Core	G06F01114	20060101		

Patents Citing This One No US, EP, or WO patent/search reports have cited this patent.



For further information, please contact:

[Technical Support](#) | [Billing](#) | [Sales](#) | [General Information](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-78918

(43) 公開日 平成10年(1998) 3月24日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/08	3 1 0	7623-5B	G 0 6 F 12/08	3 1 0 B
11/14	3 1 0		11/14	3 1 0 B
12/16	3 1 0	7623-5B	12/16	3 1 0 C

審査請求 有 請求項の数24 O L (全 33 頁)

(21) 出願番号 特願平8-234321

(22) 出願日 平成8年(1996) 9月4日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 加納 卓也

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

(72) 発明者 酒井 浩

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

(72) 発明者 林 宏雄

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

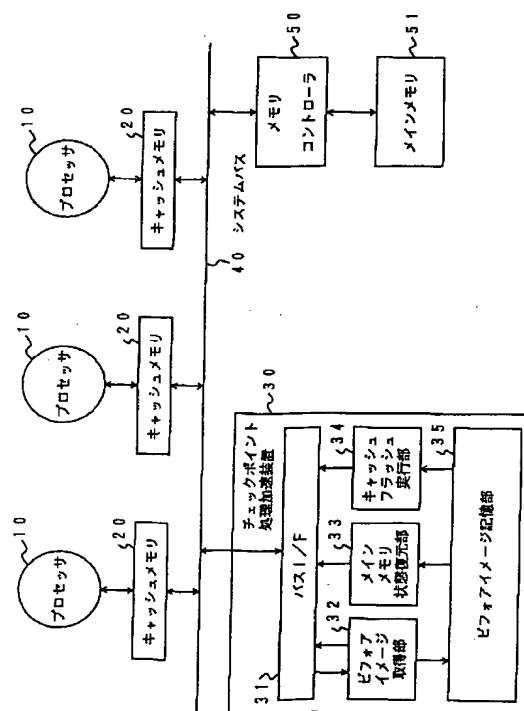
(74) 代理人 弁理士 鈴江 武彦 (外6名)

(54) 【発明の名称】 チェックポイント処理加速装置および同装置を適用した計算機システム

(57) 【要約】

【課題】 標準的なユニプロセッサなどを搭載する計算機でチェックポイント／リカバリ機能を容易に実現可能とするチェックポイント処理加速装置。

【解決手段】 キャッシュメモリ20上でのデータ更新が行なわれると、ビフォアイメージ取得部32は、その更新アドレスと更新前データとを取得して、ビフォアイメージ記憶部35に格納する。キャッシュフラッシュ実行部34は、このビフォアイメージ記憶部35に格納されたすべてのアドレスを使用し、更新状態のキャッシュブロックの内容をメインメモリ51に書き戻すことを要求するコマンドをシステムバス40に発行する。一方、メインメモリ状態復元部33は、ビフォアイメージ記憶部35に格納されたすべての更新前データについて、新しく格納したものから順に、メインメモリ51への更新書き込みを要求するコマンドをシステムバス40に発行する。



【特許請求の範囲】

【請求項1】 バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、前記プロセッサと前記メインメモリとを接続するシステムバスとを具備してなる計算機に適用されるチェックポイント処理加速装置であって、

データ更新が発生したアドレスおよびその更新前のデータを組としたビフォアイメージを複数個記憶するビフォアイメージ記憶手段と、

前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答を前記システムバスの監視によって検知し、それらが更新アドレスおよび更新前データを含む場合に、そのアドレスおよび更新前データを前記ビフォアイメージ記憶手段に格納し、更新の発生したアドレスを含むが更新前のデータを含まない場合に、その更新前のデータを読み出すためのコマンドを前記コマンドに含まれるアドレスを用いて前記システムバスに発行し、前記アドレスおよび読み出された更新前のデータを前記ビフォアイメージ記憶手段に格納するビフォアイメージ取得手段と、

前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスで示される更新状態のデータをメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行するキャッシュフラッシュ実行手段と、前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべての更新前データについて、新しく格納したものから順にメインメモリへの更新書き込みを要求するコマンドを前記システムバスに発行するメインメモリ状態復元手段とを具備してなることを特徴とするチェックポイント処理加速装置。

【請求項2】 前記メインメモリを前記キャッシュメモリ内に確保されるキャッシュブロックの大きさで分割し、この分割によって定義される複数の区画の中の少なくとも一つ以上の区画それぞれに対応させて、オンとオフとの二つの状態が割り当てられるフラグメモリをさらに設け、

前記ビフォアイメージ取得手段は、前記ビフォアイメージを取得した際、その更新アドレスに対するフラグメモリが設けられ、かつオフのときには、前記フラグメモリをオンに設定するとともに前記ビフォアイメージを前記ビフォアイメージ記憶手段に格納し、更新アドレスに対するフラグメモリが設けられ、かつオンのときには、前記ビフォアイメージの前記ビフォアイメージ記憶手段への格納を行わず、更新アドレスに対するフラグメモリが設けられていないときには、前記ビフォアイメージをビフォアイメージ記憶手段に格納する手段を具備し、前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段からビフォアイメージを取り出して、更

新状態にあるキャッシュブロックの内容の前記メインメモリへの書き戻しを要求するコマンドを発行する際、そのアドレスに対応するフラグメモリが設けられているときには、そのフラグメモリをオフに設定する手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項3】 互いに共通部分をもたないように設定される一つ以上のアドレス範囲それぞれに対応して設けられ、前記アドレス範囲に属する更新状態のキャッシュブロック数を保持するブロックカウンタと、前記システムバス上のコマンドおよびその応答を監視して、前記キャッシュブロックが更新状態に移行することを検知した際、そのアドレスに対応した前記ブロックカウンタが存在するときは、そのブロックカウンタをインクリメントし、前記キャッシュブロックが更新状態からそれ以外の状態に移行すること検知した際、そのアドレスに対応した前記ブロックカウンタがあるときは、そのブロックカウンタをデクリメントするカウンタ制御手段とをさらに具備し、

前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスに対応する前記ブロックカウンタがないか、またはその値が初期値でないときに、そのアドレスの更新状態のキャッシュブロックの内容をメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行する手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項4】 前記キャッシュメモリは、ダイレクトマップで構成され、前記ブロックカウンタは、キャッシュブロックと一対に設けられることを特徴とする請求項3記載のチェックポイント処理加速装置。

【請求項5】 前記キャッシュメモリは、nウェイのセットアソシアティブで構成され、前記ブロックカウンタは、n個のキャッシュブロックで形成されるグループと一対に設けられることを特徴とする請求項3記載のチェックポイント処理加速装置。

【請求項6】 前記メインメモリを前記キャッシュメモリ内に確保されるキャッシュブロックの大きさで分割し、この分割によって定義される複数の区画の中の少なくとも一つ以上の区画それぞれに対応させて、オンとオフとの二つの状態が割り当てられるフラグメモリをさらに設け、

前記システムバス上のコマンドおよびその応答を監視して、前記キャッシュブロックが更新状態に移行することを検知した際、そのアドレスに対応した前記フラグメモリが設けられているときは、そのフラグメモリをオンに設定し、前記キャッシュブロックが更新状態からそれ以外の状態に移行すること検知した際、そのアドレスに対応した前記フラグメモリが設けられているときは、そのフラグメモリをオフに設定するフラグメモリ制御手段を

さらに具備し、

前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスに対応する前記フラグメモリが設けられていないか、またはその状態がオンのときに、そのアドレスの更新状態のキャッシュブロックの内容をメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行する手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項7】 前記ビフォアイメージ取得手段が前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドまたはそれに対する応答を検知したときに、そのビフォアイメージを前記ビフォアイメージ記憶手段に格納するか否かその更新アドレスにより判断するアドレス判定手段をさらに具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項8】 前記ビフォアイメージ記憶手段に格納された前記ビフォアイメージそれぞれについて、前記メインメモリ状態復元手段により前記メインメモリに書き戻すか否かその更新アドレスにより判断するアドレス判定手段をさらに具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項9】 前記キャッシュメモリ上でのデータ更新が行なわれた際、その更新対象のデータが他のキャッシュメモリ上で更新されて保持されているデータであるときに、前記ビフォアイメージ取得手段は、前記ビフォアイメージを前記ビフォアイメージ記憶手段に格納しないことを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項10】 前記キャッシュブロックに割り当てられる複数の管理状態の中に、そのキャッシュブロックが保持しているデータは他のプロセッサのキャッシュメモリには保持されておらず、かつメインメモリ上と同じデータであるといったクリーンエクスクルーシブ状態を含むときに、いずれかのキャッシュメモリの発行したメインメモリデータの読み込み要求コマンドに対し、そのデータを保持していると応答する応答手段をさらに具備し、前記キャッシュブロックがクリーンエクスクルーシブ状態となるのを防ぐことを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項11】 前記ビフォアイメージ取得手段は、前記キャッシュメモリの機構を介さずに前記メインメモリの内容を更新するコマンドを検知したときに、そのコマンドの実行を中止させ、前記コマンドに含まれる更新アドレスを用いて前記メインメモリから更新前のデータを読み出すためのコマンドを前記システムバスに発行し、前記アドレスと読み出され更新前のデータとを前記ビフ

ォアイメージ格納手段に格納する手段と、前記実行を中止させたコマンドが再度発行されたときには、そのコマンドの実行の中止を行なわない手段とを具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項12】 前記ビフォアイメージ取得手段は、前記プロセッサから指示されたアドレス範囲について、前記メインメモリから更新前のデータを読み出すためのコマンドを前記システムバスに発行し、前記アドレスとその読み出された更新前のデータを組としたビフォアイメージを前記ビフォアイメージ格納手段に格納する手段を具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項13】 前記ビフォアイメージ取得手段およびキャッシュフラッシュ実行手段は、並行して実行可能であることを特徴とする請求項1、2または3記載のチェックポイント処理加速装置。

【請求項14】 前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段に格納されたビフォアイメージの中の最も早い時点で取得されたものから順に処理することを特徴とする請求項13記載のチェックポイント処理加速装置。

【請求項15】 前記ビフォアイメージ記憶手段は、互いに独立した複数のメモリで構成されることを特徴とする請求項13記載のチェックポイント処理加速装置。

【請求項16】 前記ビフォアイメージ記憶手段は、互いに独立した第1および第2のメモリを含んで構成され、

前記ビフォアイメージ取得手段は、前記キャッシュフラッシュ実行手段が動作していないときには、前記取得したビフォアイメージを前記第1のメモリに格納し、前記キャッシュフラッシュ実行手段が動作を開始したときには、前記取得したビフォアイメージを前記第2のメモリに格納する手段を具備し、

前記キャッシュフラッシュ実行手段は、前記第1のメモリに格納されたビフォアイメージから処理する手段を具備してなることを特徴とする請求項13記載のチェックポイント処理加速装置。

【請求項17】 前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ取得手段によって前記第1のメモリのすべてに前記ビフォアイメージが格納されたときに起動されることを特徴とする請求項16記載のチェックポイント処理加速装置。

【請求項18】 前記ビフォアイメージ記憶手段の残容量が予め定められた量を下回ったことを前記プロセッサに通知する通知手段をさらに具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項19】 前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ取得手段が実行中であるとき

に、更新状態のキャッシュブロックの内容を前記メインメモリへ書き戻すことを要求するコマンドの発行頻度を予め定められた値以内に抑制し、前記ビフォアイメージ取得手段が停止したときに、前記コマンドの発行頻度を高くする手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項20】 バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、前記プロセッサと前記メインメモリとを接続するシステムバスを具備し、中断した処理を再開するためのチェックポイントを定期的に採取する作成しながらデータ処理を行なう計算機システムであって、請求項1乃至19記載のいずれかのチェックポイント処理加速装置と、前記チェックポイント処理加速装置のビフォアイメージ取得手段を動作させながら通常のデータ処理を実行するデータ処理手段と、すべてのプロセッサが同期して実行するデータ処理中のコンテキストのメインメモリへの格納、および前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段が実行するすべての更新状態のブロックの内容の前記メインメモリへの書き戻しを含むチェックポイント作成手段と、故障が発生したときに、すべてのキャッシュブロックを無効化し、前記チェックポイント処理加速装置のメインメモリ復元手段により直前に採取したチェックポイント時点の状態へのメインメモリを復元し、この復元されたメインメモリに格納されている情報を用いて、データ処理を再開するロールバック／リカバリ手段とを具備してなることを特徴とする計算機システム。

【請求項21】 前記キャッシュフラッシュ起動手段の起動をチェックポイント取得手段に先行するデータ処理手段で行なうことを特徴とする請求項20記載の計算機システム。

【請求項22】 前記チェックポイント作成手段は、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段とキャッシュ操作命令の実行によるキャッシュフラッシュとを併用することを特徴とする請求項21記載の計算機システム。

【請求項23】 前記チェックポイント作成手段は、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段を停止して、キャッシュ操作命令の実行によるキャッシュフラッシュを実行することを特徴とする請求項21記載の計算機システム。

【請求項24】 請求項2記載のチェックポイント処理加速装置を適用した計算機システムであって、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段は、途中で取得したビフォアイメージから使用し始めることを特徴とする請求項23記載の計算機システム。

ム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、たとえばデータの一貫性を保証するためにスヌープ機構を有してなるキャッシュメモリを備えた計算機に適用して好適なチェックポイント加速装置および同装置を適用したチェックポイント／リカバリ方式の計算機システムに関する。

【0002】

【従来の技術】 計算機システムを高信頼化する方式の一つとして、通常のデータ処理の途中で定期的にチェックポイントをメインメモリ上に作成しておき、計算機に何らかの故障が発生したときに、直前のチェックポイントに戻って処理を再開するといったものが存在する。そして、このような方式は、チェックポイント／リカバリ方式と称され、下記の3つの種類に大別される。

【0003】 (1) 主にデータベース管理システムなどで使用され、計算機がダウンしたときに、データ自体の喪失を防いだり、データ間の論理的な整合性を維持したり、他の計算機がデータベースを引継いたりするためのもの。

【0004】 (2) ひとつのアプリケーションプログラムをプライマリプロセスとシャドウプロセスとして別の計算機に配置して実行し、プライマリプロセスを実行中の計算機がダウンしたときに、シャドウプロセスが整合性のとれた状態で引継ぐためのもの。ユーザからは、アプリケーションプログラムの実行が途中で打ち切られずに続いているように見える。

【0005】 (3) 計算機に何らかの故障が発生したときに、その計算機がダウンすることを回避するためのもの。ユーザおよびアプリケーションプログラムからは、故障が発生しなかったときと比べてほとんど違いが見えない。

【0006】 このうち、(3)のチェックポイント／リカバリ方式では、何らかの故障が発生したときに、直前のチェックポイントから処理が再開されるので、チェックポイント自身は故障発生によっても破壊されない記憶装置に格納することが必須である。そのような記憶装置は安定記憶と呼ばれ、たとえば二重化された主記憶装置などが使用される。

【0007】 このようなチェックポイント／リカバリ方式の計算機では、図29に示すように、各プロセッサは、定期的に通常のデータ処理を一時中断してチェックポイント処理を行ない(1)、それが終了すると再び通常のデータ処理を中断した時点から再開する(2)。そして、何らかの故障が発生すると(3)、プロセッサはリカバリ処理を行なう。このリカバリ処理では、メインメモリを(2)の状態に戻した後に、通常のデータ処理を再開する(4)。すなわち、(4)において(2)と同じ状態から再開することにより、障害発生の影響を受

けることなくデータ処理を続行することを目指すものである。

【0008】次に、このチェックポイント／リカバリ方式について、キャッシュメモリ、メインメモリおよびチェックポイントの関係を下記に示す。

【0009】（通常のデータ処理）障害発生時にメインメモリをチェックポイント時の状態に戻すため、メインメモリが書き換えられる場合、何らかの対策をとる必要がある。

【0010】（チェックポイント処理）キャッシュメモリに保持されている更新されたデータをすべてメインメモリに書き戻す。

【0011】（障害発生時の復旧処理）少なくとも直前のチェックポイント処理以後にメインメモリ上で更新の発生した部分をチェックポイント時点の状態に戻す必要がある。

【0012】チェックポイント／リカバリ方式を採用したフォールトトレラント計算機は、たとえばPhilip A. Bernstein, "Sequoia: A Fault-Tolerant Tightly Coupled Multiprocessor for Transaction Processing", IEEE Computer, Vol. 21, No. 2, 1988などに開示されている。

【0013】このSequoia計算機においては、通常のデータ処理の期間中、プロセッサがデータを更新すると、そのデータはキャッシュメモリ上に保持され、決してメインメモリに書き戻されることはない。そして、チェックポイント処理を開始すると、キャッシュメモリに保持されていた更新されたデータがメインメモリに書き戻される。そして、計算機に何らかの故障が発生した場合には、キャッシュメモリを無効化することにより、直前のチェックポイントの時点のメインメモリの状態から通常のデータ処理を再開することができる。これを前述したキャッシュメモリ、メインメモリおよびチェックポイントの関係に対応させると、下記のように示すことができる。

【0014】（通常のデータ処理）プロセッサにより更新されたデータは、チェックポイント処理を始めるまではメインメモリに書き戻さない。

【0015】（チェックポイント処理）キャッシュメモリに保持されている更新されたデータは、すべてメインメモリに書き戻す。

【0016】（障害発生時の復旧処理）キャッシュメモリを無効化するだけで良い。

【0017】また、このSequoia計算機では、チェックポイント／リカバリ方式を実現するために、特別なキャッシュメモリを備えている。というのは、通常のライトスルー型、あるいはコピーバック型のキャッシュメモリを使用すると、「通常のデータ処理の期間中にプ

ロセッサにより更新されたデータは、チェックポイント処理を始めるまではメインメモリに書き戻さない」という制御ができないためである。

【0018】また、USP 4, 740, 969 "Method & Apparatus for Recovering from Hardware Faults" には、下記的方式が開示されている。

【0019】（通常のデータ処理）メインメモリからキャッシュメモリにデータをロードする際、そのアドレスとデータの内容を保持する。

【0020】（チェックポイント処理）言及されていない。

【0021】（障害発生時の復旧処理）前記アドレスとデータとを用いて、メインメモリをチェックポイント時点の状態に戻す。

【0022】

【発明が解決しようとする課題】このように、Sequoia計算機の場合には、チェックポイント／リカバリのために特別なキャッシュメモリを備える必要があり、プロセッサの急激な技術改革に追従させることが難しいという問題がある。

【0023】また、USP 4, 740, 969に開示される方式では、通常のデータ処理において取得するデータの量が多くなること、およびチェックポイント処理時にキャッシュフラッシュを実行する手段を設ける必要があるという問題がある。

【0024】この発明は、このような実情に鑑みてなされたものであり、特別なキャッシュメモリを必要とせず、標準的なユニプロセッサあるいはマルチプロセッサを搭載する計算機でチェックポイント／リカバリ機能を容易に実現可能とするチェックポイント処理加速装置およびチェックポイント／リカバリ方式を提供することを目的とする。これにより、標準的な計算機における性能向上の恩恵を享受することができ、また、USP 4, 740, 969に開示される方法と比較して、通常のデータ処理において取得するデータの量がより少なく、チェックポイント処理時のキャッシュフラッシュ機能も実現するチェックポイント処理加速装置およびチェックポイント／リカバリ方式を提供することが可能となる。

【0025】

【課題を解決するための手段】この発明のチェックポイント処理加速装置は、バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、前記プロセッサと前記メインメモリとを接続するシステムバスとを具備してなる計算機に適用されるチェックポイント処理加速装置であって、データ更新の発生したアドレスおよびその更新前のデータを組としたビフォアイメージを複数個記憶するビフォアイメージ記憶手段と、前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよ

びそれに対する応答を前記システムバスの監視によって検知し、それらが更新アドレスおよび更新前データを含む場合に、そのアドレスおよび更新前データを前記ビフォアイメージ記憶手段に格納し、更新の発生したアドレスを含むが更新前のデータを含まない場合に、その更新前のデータを読み出すためのコマンドを前記コマンドに含まれるアドレスを用いて前記システムバスに発行し、前記アドレスおよび読み出された更新前のデータを前記ビフォアイメージ記憶手段に格納するビフォアイメージ取得手段と、前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスで示される更新状態のデータをメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行するキャッシュフラッシュ実行手段と、前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべての更新前データについて、新しく格納したものから順にメインメモリへの更新書き込みを要求するコマンドを前記システムバスに発行するメインメモリ状態復元手段とを具備してなることを特徴とする。

【0026】この発明のチェックポイント処理加速装置においては、プロセッサがキャッシュメモリ上でデータ更新を行なうと、ビフォアイメージ取得手段が、その更新アドレスと更新前データとを取得してビフォアイメージ記憶手段に格納する。キャッシュフラッシュ実行手段は、プロセッサからの指示に基づき、ビフォアイメージ記憶手段に格納されているすべてのアドレスを使用して、そのアドレスを有する更新状態のキャッシュブロックの内容をメインメモリに書き戻すことを要求するコマンドをシステムバスに発行することにより、キャッシュフラッシュを実行する。一方、メインメモリ状態復元手段は、プロセッサからの指示に基づき、ビフォアイメージ記憶手段に格納されているすべての更新前データを新しく格納したものから順にメインメモリの更新書き込みを要求するコマンドをシステムバスに発行することにより、メインメモリをビフォアイメージ記憶手段には何も記憶されていなかったときの状態にする。

【0027】すなわち、この発明のチェックポイント処理加速装置を適用すれば、特別なキャッシュメモリを不要とし、かつ、従来のように、プロセッサがキャッシュフラッシュ用のソフトウェアを実行することによってキャッシュフラッシュを実施することと比較して、大幅な高速化が期待される。したがって、チェックポイント処理に費やされる作業のほとんどがキャッシュフラッシュであることを考慮すれば、チェックポイント／リカバリ方式の計算機全体の性能を向上させることができる。

【0028】また、連続するチェックポイントの間でプロセッサが同じアドレスのデータを複数回更新することも十分考えられる。そこで、メインメモリをキャッシュメモリ上に確保されるキャッシュブロックの大きさで分

割し、この分割によって定義される区画に対応させて、オンとオフとの二つの状態が割り当てられるフラグメモリを設けることが好ましい。そして、キャッシュブロック内のデータが更新された際、そのアドレスに対応したフラグメモリがオフであった場合には、その更新前のデータをビフォアイメージ記憶手段に格納するとともに、そのフラグメモリをオンにする。また、プロセッサにより再度そのデータが更新された場合には、そのフラグメモリはすでにオンとなっているので、更新前のデータは格納済みであることが認識できるため、更新前のデータのビフォアイメージ記憶手段への格納は行なわない。これにより、不要な動作を削減することができるため、その性能をさらに向上させることが可能となる。このフラグメモリは、必ずしもすべての区画に対応させて設けなければならないものではなく、一部の区画に対応させて設けた場合であっても有効である。

【0029】また、互いに共通部分をもたないようにアドレス範囲を設定し、この設定したアドレス範囲それぞれに対応させてブロックカウンタを設けることが好ましい。この場合、このブロックカウンタを、その対応するアドレス範囲に含まれるキャッシュブロックの中で更新状態のキャッシュブロックがいくつあるかを示すように管理する。したがって、このブロックカウンタが初期値（たとえば0）を示す場合、このアドレス範囲に関しては、キャッシュフラッシュが不要であることが即座に認識できるため、チェックポイント処理の所要時間を大幅に短縮することが可能となる。

【0030】また、この発明の計算機システムは、バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、前記プロセッサと前記メインメモリとを接続するシステムバスを具備し、中断した処理を再開するためのチェックポイントを定期的に採取する作成しながらデータ処理を行なう計算機システムであって、請求項1乃至19記載のいずれかのチェックポイント処理加速装置と、前記チェックポイント処理加速装置のビフォアイメージ取得手段を動作させながら通常の前記データ処理を実行するデータ処理手段と、すべてのプロセッサが同期して実行するデータ処理中のコンテキストのメインメモリへの格納、および前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段が実行するすべての更新状態のブロックの内容の前記メインメモリへの書き戻しを含むチェックポイント作成手段と、故障が発生したときに、すべてのキャッシュブロックを無効化し、前記チェックポイント処理加速装置のメインメモリ復元手段により直前に採取したチェックポイント時点の状態へのメインメモリを復元し、この復元されたメインメモリに格納されている情報を用いて、データ処理を再開するロールバック／リカバリ手段とを具備してなることを特徴とする。

【0031】中断した処理を再開するためのチェックポイントを定期的に採取する計算機システムにおいては、このチェックポイントの採取に費やす時間をいかに短縮するかが重要である。すなわち、このような計算機システムに、前述したようなチェックポイント処理加速装置を適用すれば、システム全体の性能を向上させることが可能となる。

【0032】

【発明の実施の形態】以下、図面を参照してこの発明の実施の形態を説明する。

【0033】（第1の実施形態）図1には、この発明の第1の実施形態に係る計算機システムのシステム構成が示されている。各プロセッサ10は、キャッシュ一貫性を備えたコピーバック型のキャッシュメモリ20を有する。ただし、キャッシュメモリがプライマリとセカンダリとの2階層あるいはそれ以上の階層で構成される場合、本実施形態でいうキャッシュメモリ20とは、最もシステムバス40に近いキャッシュメモリを指す。また、キャッシュメモリによっては、命令を格納するキャッシュメモリと、データを格納するキャッシュメモリとが分かれているものもあるが、その場合、本実施形態でいうキャッシュメモリ20とは、データを格納するためのキャッシュメモリを指す。また、本システムにおけるシステムバス40上でのアドレスは、32ビットとして説明する。

【0034】まず、このチェックポイント処理加速装置30の機能について、本装置を制御するソフトウェアと関連させて説明する。

【0035】（初期化）プロセッサ10は、ある適当な時点でModified状態（後述）のキャッシュブロックが存在しない状態を作る。

【0036】（通常データ処理時）プロセッサ10は、通常データ処理を開始する時点でビフォアイメージ取得部32を起動する。プロセッサ10が、アプリケーションプログラムなどを実行している間、チェックポイント処理加速装置30は、システムバス40を監視し、プロセッサ10がキャッシュメモリ20上でデータを更新したことを検知すると、更新アドレスと更新前のデータとを記憶する。

【0037】（チェックポイント処理）プロセッサ10は、通常データ処理を一定時間、たとえば数ミリ秒～数10ミリ秒実行すると、チェックポイント処理を行なう。チェックポイント処理では、直前まで実行していたアプリケーションプログラムのコンテキスト（プロセッサのレジスタ内容など）をメインメモリ51に格納するとともに、すべてのModified状態のキャッシュブロックの内容をメインメモリ51に格納する（以下、「キャッシュフラッシュ」という）。このチェックポイント処理加速装置30は、すべてのModified状態のキャッシュブロックに対し、その内容のメインメモ

リ51への書き戻しを要求するコマンドをシステムバス40に発行することによってキャッシュフラッシュを実現する。この処理は、プロセッサ10がキャッシュフラッシュ用のソフトウェアを実行するのと比較して高速であることが期待される。チェックポイント処理が終了すると、プロセッサ10は通常データ処理を再開する。

【0038】（障害発生時の復旧処理）計算機システムに何らかの障害が発生した場合、プロセッサ10は、障害箇所を同定し、必要であれば障害の発生した装置を切り離すとともに、すべてのキャッシュブロック20を無効化する。そして、チェックポイント処理加速装置30は、その内部に記憶されているビフォアイメージを使って、メインメモリ51に更新前のデータを書き戻すコマンドを発行することにより、メインメモリ51の状態を直前のチェックポイント処理時の状態に戻す。その後、プロセッサ10は通常データ処理を再開する。

【0039】すなわち、本実施形態のチェックポイント処理加速装置30は、下記の3つの機能を有する。

【0040】（1）通常データ処理時
プロセッサ10がデータ更新を行なったことを検知し、そのビフォアイメージを取得して記憶する。

【0041】（2）チェックポイント処理
プロセッサ10からの指示により、記憶していたビフォアイメージを使ってModified状態のキャッシュブロックの内容のメインメモリ51への書き戻しを要求するコマンドを発行する。

【0042】（3）障害発生時の復旧処理
プロセッサ10からの指示により、記憶していたビフォアイメージを使ってメインメモリ51に更新前のデータを書き戻すコマンドを発行する。

【0043】なお、ビフォアイメージを記憶するのは、（2）および（3）の機能を実現するためであるから、必ずしもすべてのデータ更新について、そのビフォアイメージを取得する必要はない。たとえば、チェックポイントの後、あるアドレスに対して最初に更新が発生した場合、ビフォアイメージの取得は必須であるが、2度目以降の場合には、ビフォアイメージの取得は不要である。

【0044】このチェックポイント処理加速装置30は、システムバス40上のコマンドとその応答とを監視し、あるいは能動的にコマンドを発行することにより処理を行なうので、ここでキャッシュメモリ20およびシステムバス40の構造、ならびにコマンドとそれに対する応答とについて説明する。ただし、ここで述べるものは、本実施形態のメモリ状態復元部33を適用可能な計算機の一例である。たとえば、下記に示すキャッシュブロックの取り得る状態やシステムバス40上のコマンドや応答には、それぞれ数種類のバリエーションが存在するが、いずれについても本実施形態の主旨にしたがったチェックポイント処理加速装置30が構成可能である。

【0045】（システムバスの構造）システムバス40は、アドレス／データ線の他、制御用信号線として、コマンド信号線、モディファイド応答信号線、およびシェアード応答信号線を含む。アドレス／データ信号線は、コマンドのアドレスおよびデータを送るための信号線であり、共通の信号線を時分割で使う方式と、アドレス／データそれぞれに独立した信号線を設ける方式とがあり、いずれであっても構わない。コマンド信号線は、コマンドの種類を示す他、コマンドに関する各種情報を送るための信号線である。シェアード応答信号線は、他のプロセッサ／キャッシュメモリにより発行されたコマンドに対して、ブロックを共有していることを通知するために用いられる。そして、モディファイド応答信号線は、他のプロセッサ／キャッシュメモリにより発行されたコマンドに対して、ブロックの更新データを保持していることを通知するために用いられる。

【0046】（キャッシュブロックの状態）キャッシュブロックの取り得る状態は、下記の通りである。

【0047】（1）Invalid状態

そのキャッシュブロックには有効なデータが存在しない。

【0048】（2）Modified状態

そのキャッシュブロックには更新されたデータが存在する。また、他のプロセッサ10のキャッシュブロックには、このデータは存在しない。

【0049】（3）Clean-Shared状態

そのキャッシュブロックにはメインメモリ51と同一内容のデータが存在する。また、他のプロセッサ10のキャッシュブロックにも、このデータが存在する可能性がある。

【0050】（4）Clean-Exclusive状態

そのキャッシュブロックにはメインメモリ51と同一内容のデータが存在する。また、他のキャッシュメモリ20のキャッシュブロックには、このデータは存在しない。

【0051】（コマンドおよび応答の仕様）システムバス40上のコマンドおよびそれに対するキャッシュメモリ20あるいはメインメモリ51からの応答は、次の通りである。ただし、このうち、Read-Line-Non-Coherentは、システムバス40ではサポートされているが、キャッシュメモリ20が発行するコマンドではない。

【0052】（1）Read-Line（キャッシュブロックデータのリード要求）

発行される状況…プロセッサ10がデータを読込もうとして、キャッシュメモリ20中にそのデータが無い場合。

【0053】応答（ケース1）…Modified状態のキャッシュブロックを有するキャッシュメモリ20が

あれば、そのキャッシュメモリ20は、モディファイド応答信号線をアサートして、Read-Lineコマンドを発行したキャッシュメモリ20とメインメモリ51にデータが転送される。両方のキャッシュブロックの状態は、Clean-Sharedとなる。

【0054】応答（ケース2）…Modified状態のキャッシュブロックを有するキャッシュメモリ20はないが、Clean-SharedまたはClean-Exclusive状態のキャッシュブロックを有するキャッシュメモリ20があれば、そのキャッシュメモリ20は、シェアード応答信号線をアサートする。メインメモリ51またはそのデータを保持するキャッシュメモリ20がRead-Lineコマンドを発行したキャッシュメモリ20にデータを転送する。いずれがデータ転送するかは、計算機に依存する。両方のキャッシュブロックの状態は、Clean-Sharedとなる。

【0055】応答（ケース3）…前記いずれでもない場合、モディファイド応答信号線もシェアード応答信号線もアサートされず、メインメモリ51からRead-Lineコマンドを発行したキャッシュメモリ20にデータが転送される。キャッシュブロックの状態は、Clean-Exclusiveとなる。

【0056】（2）Read-Line-with-Invalidate（キャッシュブロックデータのリードと、他のキャッシュブロックの無効化要求）

発行される状況…プロセッサ10がデータを更新しようとして、キャッシュメモリ20中にそのデータが無い場合。

【0057】応答（ケース1）…Modified状態のキャッシュブロックを有するキャッシュメモリ20があれば、そのキャッシュメモリ20は、モディファイド応答信号線をアサートして、Read-Line-with-Invalidateコマンドを発行したキャッシュメモリ20とメインメモリ51にデータが転送される。計算機によっては、Modified状態のキャッシュブロックを有するキャッシュメモリ20がRetryを応答して、メインメモリ51にデータを書戻し、Read-Line-with-Invalidateコマンドを発行したキャッシュメモリ20が再度同じコマンドを発行すると、今度はメインメモリ51が応答することもある（後述するケース3）。本実施形態では、前者であるとして説明する。要求を発行した側のキャッシュブロックの状態は、Modifiedとなり、応答した側は、Invalid状態となる。

【0058】応答（ケース2）…Clean-SharedまたはClean-Exclusive状態のキャッシュブロックを有するキャッシュメモリ20があれば、そのキャッシュメモリ20は、シェアード応答信号線をアサートする。メインメモリ51またはそのデータを保持するキャッシュメモリ20がRead-Line

ーwith-Invalidateコマンドを発行したキャッシュメモリ20にデータを転送する。いずれがデータ転送するかは、計算機に依存する。要求を発行した側のキャッシュブロックの状態は、Modifiedとなり、応答した側は、Invalid状態となる。

【0059】 応答（ケース3）…前記いずれでもない場合、モディファイド応答信号線もシェアード応答信号線もアサートされず、メインメモリ51からRead-Line-with-Invalidateコマンドを発行したキャッシュメモリ20にデータが転送される。キャッシュブロックの状態は、Modifiedとなる。

【0060】 (3) Write-Line（キャッシュブロックデータのメインメモリへの書き込み）
発行される状況…Modified状態のキャッシュブロックに他のアドレスのデータを格納しなければならなくなったため、Modified状態のキャッシュブロックの内容をメインメモリ51に書き戻す場合。

【0061】 応答…メインメモリ51がこのコマンドに含まれるデータにより更新される。キャッシュブロックの内容は、Clean-Sharedとなる。

【0062】 (4) Invalidate（他のキャッシュメモリのデータ無効化要求。アドレスの転送のみ）
発行される状況…プロセッサ10がClean-Shared状態のキャッシュブロックのデータを更新しようとする場合。

【0063】 応答…本コマンドを発行した側は、Modified状態となり、受信した側は、Invalid状態となる。

【0064】 (5) Read-Line-Non-Coherent（メインメモリデータのリード要求）
発行される状況…メインメモリ51からI/O装置へのDMA転送等で使用応答…キャッシュメモリ20の状態に関わらず、常にメインメモリ51が要求されたアドレスの内容を応答として返す。キャッシュブロックの状態は不変である。

【0065】 以上、本実施形態のキャッシュメモリ20およびシステムバス40などを説明した。

【0066】（チェックポイント処理加速装置の構成）
チェックポイント処理加速装置30は、ビフォアイメージ記憶部35、ビフォアイメージ取得部32、キャッシュフラッシュ実行部34およびメインメモリ状態復元部33で構成される。

【0067】 ビフォアイメージ記憶部35は、図2に示すように、更新アドレスおよび更新前データの組（ビフォアイメージ）を記憶する複数のエン트리と、キャッシュフラッシュ実行部34が使用するFポインタと、ビフォアイメージ取得部32およびメインメモリ状態復元部33が使用するWポインタとで構成される。

【0068】 ビフォアイメージ記憶部35に記憶する更新前のデータは、キャッシュブロックと同じ大きさ（こ

こでは、Bビットとする）である。これは、キャッシュメモリ20間、あるいはキャッシュメモリ20とメインメモリ51間のデータ転送の単位と同じにするためである。

【0069】 Wポインタは、ビフォアイメージ取得部32が新たに取得したビフォアイメージを格納するエントリを示す。プロセッサ10の指示により、ビフォアイメージ取得部32が起動されると、Wポインタは、最初のエントリを指すように初期化される。そして、ビフォアイメージ取得部32が新たに取得したビフォアイメージをWポインタの指すエントリに格納すると、Wポインタがインクリメントされ、その次のエントリを指すように制御される。また、プロセッサ10の指示によりメインメモリ状態復元部33が起動されると、Wポインタが次々とデクリメントされながら、Wポインタの指すエントリに格納されている更新アドレスと更新前データの組を使って、メインメモリ51の内容が更新前の状態に戻されていく。

【0070】 Fポインタは、キャッシュフラッシュ実行部34が次に処理対象とするビフォアイメージを格納しているエントリを指す。プロセッサ10の指示により、キャッシュフラッシュ実行部34が起動されると、Fポインタは最初のエントリを指すよう初期化される。そして、キャッシュフラッシュ実行部34は、Fポインタの指すエントリに格納されている更新アドレスに対して、Read-Lineコマンドを発行しては、Fポインタが次のエントリを指すよう制御される。そして、Fポインタの値とWポインタの値が一致すると、キャッシュフラッシュ実行部34は終了する。

【0071】 ビフォアイメージ取得部32は、システムバス40上のコマンドおよび応答を監視し、あるいは必要によりコマンドを発行することにより、プロセッサ10がキャッシュメモリ20上でデータを更新した際、そのビフォアイメージをビフォアイメージ記憶部35に格納する。ビフォアイメージ取得部32が前記コマンドを検知したときの動作を下記に示す。

【0072】 (1) Read-Line
シェアード応答信号線をアサートする。

【0073】 (2) Read-Line-with-Invalidate

もし、モディファイド応答信号線がアサートされれば何もしない。そうでなければ、コマンドに含まれる更新アドレスと、キャッシュメモリ20またはメインメモリ51からの応答に含まれる更新前データとをビフォアイメージ記憶部35に格納する。

【0074】 (3) Invalidate
コマンドに含まれる更新アドレスを使って、Read-Line-Non-Coherentコマンドを発行する。そして、前記更新アドレスとメインメモリ51からの応答に含まれる更新前データとをビフォアイメージ記

憶部35に格納する。

【0075】図3には、このビフォアイメージ取得部32の処理フローが示されている。

【0076】Read-Lineコマンドは、プロセッサ10がデータを更新するときに発行されるコマンドではないので、ビフォアイメージを取得することはない。しかし、シェアード応答信号線をアサートし、キャッシュブロックがClean-Exclusive状態となるのを防ぐ必要がある。というのは、プロセッサ10がClean-Exclusive状態のキャッシュブロックのデータを更新する場合、他のキャッシュメモリ20はそのデータを保持していないため、システムバス40に何のコマンドも発行されることなく、Clean-Exclusive状態からModified状態に移移する。そのため、ビフォアイメージ取得部32は、ビフォアイメージを取得できなくなる。そこで、ビフォアイメージ取得部32がシェアード応答信号線をアサートすることにより、キャッシュブロックはClean-Shared状態となり、プロセッサ10によるデータの更新では、システムバス40にInvalidateコマンドが発行されることになる。

【0077】Read-Line-with-Invalidateコマンドの場合、キャッシュメモリ20あるいはメインメモリ51から更新前のデータを含む応答があるので、ビフォアイメージ取得部32は、それらを抽出すれば良い。ただし、モディファイド応答信号線がアサートされている場合、他のキャッシュメモリ20がそのデータをModified状態で保持していたことになるので（したがって、そのアドレスに対応するビフォアイメージはすでにビフォアイメージ記憶部35に記憶されている）、その場合にはビフォアイメージを記憶しない。これにより、ビフォアイメージ記憶部32の利用効率の向上、キャッシュフラッシュ時間の短縮などが期待できる。

【0078】Invalidateコマンドの場合、本コマンドおよび応答はデータを含まないため、ビフォアイメージ取得部32が、Read-Line-Non-Coherentコマンドを発行し、メインメモリ51からの応答に含まれる更新前データを取得する。ここで、もしRead-Lineコマンドを発行すると更新されたデータを保持しているキャッシュブロックが応答するので、更新前のデータを取得することはできない。すなわち、Read-Line-Non-Coherentコマンドを発行することが必須である。

【0079】キャッシュフラッシュ実行部34は、プロセッサ10の指示により、まずFポインタが最初のエントリを指すようにした後、Fポインタの指すエントリ中の更新アドレスを使って、Read-Lineコマンドを発行し、Fポインタを進める。Read-Lineコマンドを発行するのは、もし、そのアドレスのデータを

Modified状態で保持するキャッシュブロックがあれば、そのデータをメインメモリ51にも転送し、そのキャッシュブロックの状態はModifiedからClean-Sharedに変化するというRead-Lineとその応答の性質を利用するためである。そして、FポインタがWポインタと等しくなれば、キャッシュフラッシュ処理を終了する。この時、キャッシュブロックの状態はすべてInvalidもしくはClean-Sharedとなっている。

【0080】このキャッシュフラッシュ実行部34は、ビフォアイメージ記憶部35に格納されているすべてのアドレスについて、Read-Lineコマンドを発行しているが、これには無駄がある。というのは、通常のデータ処理において、たとえばa番地のデータをModified状態で保持しているキャッシュブロックが、その後にb番地のデータを格納するため、a番地のデータをメインメモリ51に書き戻した場合、a番地のModified状態のキャッシュブロックは実在しないにもかかわらず、キャッシュフラッシュ実行部34は、a番地のビフォアイメージが存在するため、a番地に対してRead-Lineコマンドを発行することが、少なからず起こるためである。これを改善する方法については、他の実施形態において説明する。図4には、キャッシュフラッシュ実行部34の処理フローが示されている。

【0081】メインメモリ状態復元部33は、プロセッサ10の指示により、Wポインタをデクリメントし、Wポインタの指すエントリ中のビフォアイメージをWrite-Lineコマンドを使って、メインメモリに書き戻す。ここで、a番地に対するビフォアイメージが2つ以上存在する場合、チェックポイント時の状態に戻すには、より早い時点で取得したビフォアイメージをより遅い時点でメインメモリ51に書き戻す必要がある。したがって、Wポインタが現在指すエントリから最初のエントリに向かってデクリメントするのは、前記の制約を満たす最も容易な方法である。図5には、メインメモリ状態復元部33の処理フローが示されている。

【0082】次に、プロセッサ10によるチェックポイント処理加速装置30の制御方式を図6に基づいて説明する。

【0083】通常のデータ処理では、ビフォアイメージ取得部32を起動した後、アプリケーションプログラムやオペレーティングシステムを実行する。そして、適当なタイミングになると、通常のデータ処理を中断して、チェックポイント処理を開始する。

【0084】チェックポイント処理では、まず、プロセッサ10を割り込み禁止状態にして、外部機器などからの割り込みを受けつけないようにする。これは、チェックポイント処理中は、通常のデータ処理を一切実行しないようにするためである。ただし、計算機の故障発生を

通知する割り込みについては、通常の割り込みより高い優先度を設定するなど、前記割り込み禁止状態にあっても受け付けられるようになっている必要がある。

【0085】次に、プロセッサ10は、キャッシュフラッシュ実行部34を起動し、通常のデータ処理を中断した時点でのコンテキスト、すなわちレジスタの内容のメインメモリ51の然るべきアドレスへの書き込みを行なう。ただし、本実施形態ではコピーバック型のキャッシュメモリが使用されるので、この時点では通常のデータ更新と同じように、キャッシュブロックにModified状態で保持され、ビフォアイメージ記憶部35にそのアドレスのビフォアイメージが記憶されている。

【0086】その後、プロセッサ10は、キャッシュフラッシュ実行部34が、ビフォアイメージ記憶部35内のビフォアイメージを使って、すべてのModified状態のキャッシュブロックのデータがメインメモリ51に書き戻されるのを待つ。プロセッサ10のコンテキストも、このとき、メインメモリ51に書き戻される。キャッシュフラッシュ実行部34が実行を終了すると、メインメモリ51に保存したコンテキストを対応するレジスタに戻し、割り込み禁止状態を解除することにより、通常のデータ処理を再開する。

【0087】次に、チェックポイント処理加速装置30の動作を説明する。

【0088】(初期化)プロセッサ10は、ソフトウェアを実行することにより、すべてのキャッシュブロックをModified状態以外の状態にする。

【0089】(通常のデータ処理時)プロセッサ10は、通常のデータ処理を開始する時点で、ビフォアイメージ取得部32を起動する。ここでは、図7を参照し、通常のデータ処理として、プロセッサ10が、a番地のデータをA0からA1に、b番地のデータをB0からB1に、a番地のデータをA1からA2に、この順で更新する場合について説明する。ただし、a番地、b番地は、ともにキャッシュブロックBLK0に対応し、BLK0は、初期状態ではa番地のデータA0をClean-Shared状態で保持しているとする。

【0090】(1)初期状態を示す。

【0091】(2)プロセッサ10がa番地のデータをA1に更新しようとするが、キャッシュブロックBLK0がa番地のデータをClean-Shared状態で保持しているため、a番地に対するInvalidateコマンドが発行される。

【0092】(3)そのデータを有する他のキャッシュメモリ20は、a番地のキャッシュブロックがあれば無効化し、終了したことを示す応答を返す。キャッシュメモリ20上のデータが、A0からA1へと更新される。

【0093】(4)ビフォアイメージ取得部32は、a番地に対してRead-Line-Non-Coherentコマンドを発行する。

【0094】(5)メインメモリ51がデータA0を応答する。ビフォアイメージ取得部32は、a番地とデータA0の組を、ビフォアイメージ記憶部35に格納する。

【0095】(6)プロセッサ10がb番地のデータをB1に更新しようとするが、キャッシュブロックBLK0がa番地のデータA1をModified状態で保持しているため、a番地のデータ書き戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にb番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0096】(7)メインメモリ51がデータB0を含む応答を返す。この動作例は、初期化直後であるので、モディファイド応答信号線はアサートされない。このとき、ビフォアイメージ取得部32は、b番地とデータB0の組をビフォアイメージ記憶部35に格納する。

【0097】(8)ライトバッファに保持されていたa番地に対するWrite-Lineコマンドが発行される。

【0098】(9)メインメモリ51のa番地のデータがA1に変わる。

【0099】(10)プロセッサ10がa番地のデータをA2に更新しようとするが、キャッシュブロックBLK0がb番地のデータB1をModified状態を保持しているため、b番地のデータ書き戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にa番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0100】(11)メインメモリ51がデータA1を含む応答を返す。ビフォアイメージ取得部32は、a番地とデータA1の組をビフォアイメージ記憶部35に格納する。

【0101】(12)ライトバッファに保持されていたb番地に対するWrite-Lineコマンドが発行される。

【0102】(13)メインメモリ51のb番地のデータがB1に変わる。

【0103】以上、プロセッサ10、メインメモリ51およびビフォアイメージ取得部32の動作例を示した。

【0104】(チェックポイント処理時)前述した状態で、プロセッサ10がキャッシュフラッシュ実行部34を起動すると、キャッシュフラッシュ実行部34は、a番地、b番地、a番地の順にRead-Lineコマンドを発行する。その様子を図8に示す。ただし、ビフォアイメージの欄の下線は、Fポインタの指す位置を表わす。

【0105】(1)初期状態を示す。

【0106】(2)キャッシュフラッシュ実行部34は、最初に取得したビフォアイメージa番地とデータA0の組を使い、a番地に対するRead-Lineコマ

ンドを発行する。

【0107】(3) BLK0はa番地のデータA2をModified状態で保持しているので、モディファイド応答信号線をアサートし、メインメモリ51への書き戻しが行なわれ、メインメモリ51のa番地がA2に変わる。BLK0の状態は、Clean-Sharedに変化する。

【0108】(4) キャッシュフラッシュ実行部34は、ビフォアイメージb番地とデータB0の組を使い、b番地に対してRead-Lineコマンドを発行する。

【0109】(5) BLK0はa番地のデータをClean-Shared状態で保持しているので、モディファイド応答信号線もシェアード応答信号線もアサートしない。メインメモリ51がB1を応答する。BLK0の状態は変化しない。

【0110】(6) キャッシュフラッシュ実行部34は、ビフォアイメージa番地とデータA1の組を使い、a番地に対してRead-Lineコマンドを発行する。

【0111】(7) BLK0はClean-Shared状態であるので、シェアード応答信号線をアサートする。メインメモリ51がA2を応答する。BLK0の状態は変化しない。

【0112】キャッシュフラッシュ実行部34の動作が終了すると、Modified状態であったすべてのキャッシュブロックの内容がメインメモリ51に書き戻されるとともに、それらキャッシュブロックの状態がClean-Shared状態になる。これは、初期化直後と同じ状態であり、したがって、キャッシュフラッシュ実行部34が終了すると、プロセッサ10は直ちに通常のデータ処理を始めることができる。

【0113】以上、ビフォアイメージを使った、キャッシュフラッシュ実行部34の動作例を示した。

【0114】(障害発生時の復旧処理) 計算機に何らかの障害が発生した場合、プロセッサ10は、障害の箇所を同定し、必要であれば障害の発生した装置を切り離すとともに、すべてのキャッシュブロックを無効化する。そして、チェックポイント処理加速装置30のメインメモリ状態復元部33が、ビフォアイメージを使って、Write-Lineコマンドを発行することにより、メインメモリ51の状態を直前のチェックポイント処理時の状態に戻す。

【0115】この様子を、先に説明したチェックポイント処理の開始前に故障が発生した場合を例に図9を参照して説明する。

【0116】(1) 初期状態を示す。

【0117】(2) プロセッサ10がすべてのキャッシュブロックの無効化を行なう命令を実行する。

【0118】(3) この命令実行の結果、BLK0はI

nvalid状態となる。

【0119】(4) メインメモリ状態復元部33が、最後に取得したビフォアイメージa番地とデータA1の組をもとに、Write-Lineコマンドを発行する。

【0120】(5) メインメモリ51のa番地にA1が書かれるが、もともとA1であったので変わらない。

【0121】(6) メインメモリ状態復元部33が、その前に取得したビフォアイメージb番地とデータB0の組をもとに、Write-Lineコマンドを発行する。

【0122】(7) メインメモリ51のb番地がB0に変わる。

【0123】(8) メインメモリ状態復元部33が、その前に取得したビフォアイメージa番地とデータA0の組をもとに、Write-Lineコマンドを発行する。

【0124】(9) メインメモリ51のa番地がA0に変わる。

【0125】この結果、メインメモリ51はチェックポイント処理終了時点の状態に戻る。

【0126】以上、メインメモリ状態復元部33の動作例を示した。

【0127】このように、本実施形態のチェックポイント処理加速装置30を適用すれば、特別なキャッシュメモリなどを必要とせずに、標準的な構成をもつ計算機システムにおいて効率的なキャッシュフラッシュが実現され、かつ高性能なチェックポイントリカバリ方式の計算機システムが構築可能となる。

【0128】(第2の実施形態) 次に、この発明の第2の実施形態を説明する。図10には、本実施形態に係る計算機システムのシステム構成が示されている。チェックポイント処理加速装置30が適用される計算機の構成は第1の実施形態と同じである。

【0129】本実施形態のチェックポイント処理加速装置30は、ビフォアイメージ記憶部35、フラグメモリ36、ビフォアイメージ取得部32、キャッシュフラッシュ実行部34およびメインメモリ状態復元部33で構成される。

【0130】ビフォアイメージ記憶部35は、第1の実施形態のビフォアイメージ記憶部35と同一である。

【0131】フラグメモリ36は、メインメモリ51をキャッシュブロック単位ごとに1ビットの情報を保持する領域である。なお、実装されているメインメモリ51の一部分についてフラグメモリ36を備えるだけでも良い。フラグメモリ36は、そのアドレスに対するビフォアイメージがビフォアイメージ記憶部35に格納され、かつキャッシュフラッシュ実行部34が未だそのアドレスに対してRead-Lineコマンドを発行していないときにだけオンになるように制御される。したがって、あるアドレスに対するフラグメモリ36がオンになっているということは、キャッシュフラッシュ実行部34が、将来、そのアドレスに対してRead-Line

コマンドを発行することを表わしている。また、初期化終了時やチェックポイント処理終了時は、すべてのフラグメモリ36の値がオフになっている。

【0132】ビフォアイメージ取得部32は、ビフォアイメージを取得する部分に関しては第1の実施形態と同じである。しかし、前述したフラグメモリ36の参照/更新の処理が追加され、また、フラグメモリ36の値によって、ビフォアイメージを格納しないことがある。ビフォアイメージ取得部32の処理フローを図11に示す。

【0133】図11に示したように、ビフォアイメージ取得部32は、更新アドレスに対応するフラグメモリ36の値を参照し、もしオフであれば、それをオンに変更するとともに、第1の実施形態と同一の方法でビフォアイメージを取得する。一方、オンであれば、ビフォアイメージを取得・格納する必要はない。それは、下記の理由による。

【0134】(1) フラグメモリ36をオンにするのは、ビフォアイメージ取得部32がビフォアイメージを取得して、ビフォアイメージ記憶部35に格納する時だけである。したがって、フラグメモリ36がオンであるということは、そのアドレスに対するビフォアイメージが格納されているので、メインメモリ状態復元部33は正しく動作する。

【0135】(2) また、フラグメモリ36がオンということは、キャッシュフラッシュ実行部34が、そのアドレスに対して将来Read-Lineコマンドを発行することを示している。したがって、キャッシュフラッシュ実行部34にとっても、フラグメモリ36がオンの時には、ビフォアイメージを格納する必要はない。

【0136】なお、あるアドレスに対応するフラグメモリ36が無い場合には、そのアドレスに対応するビフォアイメージを常に採取・格納すれば良い。つまり、そのようなアドレスに関しては、第1実施形態の場合と同じ動作となる。

【0137】キャッシュフラッシュ実行部34は、第1の実施形態のキャッシュフラッシュ実行部34に、Read-Lineコマンドを発行する際、フラグメモリ36をオフにする機構を追加したものである。なお、あるアドレスに対応するフラグメモリ36が無い場合には、図12に示す処理フローにおいて、フラグメモリ36をオフにする部分だけを実行しないようにすればよい。

【0138】メインメモリ状態復元部33は、第1の実施形態と同じである。

【0139】ここで、チェックポイント処理加速装置30の動作を説明する。

【0140】(初期化) プロセッサ10は、すべてのフラグメモリ36をオフにする。これと同期して、プロセッサ10はソフトウェアを実行することにより、すべてのキャッシュブロックをModified状態以外の状

態にする。

【0141】(通常データ処理時) プロセッサ10は、通常データ処理を開始する時点で、ビフォアイメージ取得部32を起動する。ここでは、図13を参照し、通常データ処理として、プロセッサ10が、a番地のデータをA0からA1に、b番地のデータをB0からB1に、a番地のデータをA1からA2に、この順で更新する場合について説明する。ただし、a番地、b番地は、ともにキャッシュブロックBLK0に対応し、BLK0は、初期状態ではa番地のデータA0をClean-Shared状態で保持しているとする。

【0142】(1) 初期状態を示す。

【0143】(2) プロセッサ10がa番地のデータをA1に更新しようとするが、キャッシュブロックBLK0がa番地のデータをClean-Shared状態で保持しているため、a番地に対するInvalidateコマンドが発行される。

【0144】(3) そのデータを有する他のキャッシュメモリ20は、a番地のキャッシュブロックがあれば無効化し、終了したことを示す応答を返す。

【0145】(4) ビフォアイメージ取得部32は、a番地に対応するフラグメモリ36がオフであるため、これをオンにし、a番地に対してRead-Line-Non-Coherentコマンドを発行する。

【0146】(5) メインメモリ51がデータA0を応答する。ビフォアイメージ取得部32が、a番地とデータA0の組をビフォアイメージ記憶部35に格納する。

【0147】(6) プロセッサ10がb番地のデータをB1に更新しようとするが、キャッシュブロックBLK0がa番地のデータA1をModified状態で保持しているため、a番地のデータ書戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にb番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0148】(7) メインメモリ51がデータB0を応答する。ビフォアイメージ取得部32は、b番地に対応するフラグメモリ36がオフであるため、これをオンにし、b番地とデータB0の組をビフォアイメージ記憶部35に格納する。

【0149】(8) ライトバッファに保持されていたa番地に対するWrite-Lineコマンドが発行される。

【0150】(9) メインメモリ51のa番地がA1に変わる。

【0151】(10) プロセッサ10がa番地のデータをA2に更新しようとするが、キャッシュブロックBLK0がb番地のデータB1をModified状態で保持しているため、b番地のデータ書き戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にa番地に対するRead-Line-w

ith-Invalidateコマンドが発行される。

【0152】(11)メインメモリ51がデータA1を応答する。ビフォアイメージ取得部32は、a番地に対応するフラグメモリ36がオンであるため、ビフォアイメージを格納しない。

【0153】(12)ライトバッファに保持されていたa番地に対するWrite-Lineコマンドが発行される。

【0154】(13)メインメモリ51のa番地がA1に変わる。

【0155】以上、プロセッサ10、メインメモリ51、およびビフォアイメージ取得部32の動作例を示した。

【0156】(チェックポイント処理時)前述した状態で、プロセッサ10がキャッシュフラッシュ実行部34を起動すると、キャッシュフラッシュ実行部34は、a番地、b番地の順にRead-Lineコマンドを発行する。その様子を図14に示す。ただし、ビフォアイメージの欄の下線は、Fポインタが指す位置を表わす。

【0157】(1)初期状態を示す。

【0158】(2)キャッシュフラッシュ実行部34は、最初に、ビフォアイメージa番地とデータA0の組を使い、a番地に対応するフラグメモリ36をオフにするとともに、a番地に対してRead-Lineコマンドを発行する。

【0159】(3)BLK0はa番地のデータA2をModified状態で保持しているので、モディファイド応答信号線がアサートされ、メインメモリ51への書き戻しが行なわれ、メインメモリ51のa番地がA2に変わる。BLK0の状態は、Clean-Sharedに変化する。

【0160】(4)キャッシュフラッシュ実行部34は、ビフォアイメージb番地とデータB0の組を使い、b番地に対応するフラグメモリ36をオフにするとともに、b番地に対してRead-Lineコマンドを発行する。

【0161】(5)BLK0はa番地のデータをClean-Shared状態で保持しているので、モディファイド応答信号線もシェアード応答信号線もアサートされない。メインメモリ51がB1を応答する。BLK0の状態は変化しない。

【0162】キャッシュフラッシュ実行部34の動作が終了すると、Modified状態であったすべてのキャッシュブロックの内容がメインメモリ51に書き戻されるとともに、それらキャッシュブロックの状態がClean-Shared状態になる。また、すべてのフラグメモリ36の値がオフとなる。これは、初期化直後と同じ状態であり、したがって、キャッシュフラッシュ実行部34の動作が終了すると、プロセッサ10は直ちに通常のデータ処理を始めることができる。

【0163】以上、ビフォアイメージを使った、キャッシュフラッシュ実行部34の動作例を示した。

【0164】(障害発生時の復旧処理) 計算機に何らかの障害が発生した場合、プロセッサ10は、障害の箇所を同定し、必要であれば障害の発生した装置を切り離すとともに、すべてのキャッシュブロックを無効化する。そして、チェックポイント処理加速装置30のメインメモリ状態復元部33が、ビフォアイメージを使って、Write-Lineコマンドを発行することにより、メインメモリ51の状態を直前のチェックポイント処理時の状態に戻す。

【0165】この様子を、先に説明したチェックポイント処理の開始前に故障が発生した場合を例に図15を参照して説明する。

【0166】(1)初期状態を示す。

【0167】(2)プロセッサ10がすべてのキャッシュブロックの無効化を行なう命令を実行する。

【0168】(3)この命令実行の結果、BLK0はInvalid状態となる。

【0169】(4)メインメモリ状態復元部33が、最後に取得したビフォアイメージb番地とデータB0の組をもとに、Write-Lineコマンドを発行する。

【0170】(5)メインメモリ51のb番地がB0に変わる。

【0171】(6)メインメモリ状態復元部33が、ビフォアイメージa番地とデータA0の組をもとに、Write-Lineコマンドを発行する。

【0172】(7)メインメモリ51のa番地がA0に変わる。

【0173】(8)プロセッサ10は、フラグメモリ51をすべてオフにする。

【0174】この結果、メインメモリ51はチェックポイント処理終了時点の状態に戻る。

【0175】以上、メインメモリ状態復元部33の動作例を示した。

【0176】なお、フラグメモリ36は、すべてのアドレスに対して設定する必要は必ずしもない。あるアドレスについて、対応するフラグメモリ36がない場合、ビフォアイメージ取得部32は、そのアドレスに対するビフォアイメージを取得し、ビフォアイメージ記憶部35に格納するようにすれば良い。

【0177】このように、本実施形態のチェックポイント処理加速装置30では、フラグメモリ36を設けることにより、同じアドレスに対するビフォアイメージが2回以上ビフォアイメージ記憶部35に格納されることを回避できる。また、キャッシュフラッシュ実行部34が実行を終了したときには、すべてのフラグメモリ36がオフになっているため、プロセッサ10は直ちに通常のデータ処理を開始できる。

【0178】(第3の実施形態)次に、この発明の第3

の実施形態を説明する。図16には、本実施形態に係る計算機システムのシステム構成が示されている。チェックポイント処理加速装置30が適用される計算機の構成は第1の実施形態と同じである。

【0179】本実施形態のチェックポイント処理加速装置30は、ビフォアイメージ記憶部35、ブロックカウンタ38、ブロックカウンタ制御部37、ビフォアイメージ取得部32、キャッシュフラッシュ実行部34およびメインメモリ状態復元部33で構成される。

【0180】ビフォアイメージ記憶部35は、第1の実施形態のビフォアイメージ記憶部35と同一である。

【0181】ブロックカウンタ38は、1つ以上のアドレス範囲の各々に対して設けられたカウンタの集合であり、各アドレス範囲に含まれるModified状態のキャッシュブロックの個数を保持するのに用いられる。ただし、各アドレス範囲は、互い共通部分を持たないように設定する。このアドレス範囲の設定の一例を、キャッシュメモリ20がダイレクトマップ方式の場合について説明する。

【0182】ひとつのキャッシュブロックに対応するアドレスの集合をひとつのアドレス範囲とする。この場合、ブロックカウンタに属するひとつのカウンタは、対応するキャッシュブロックでModified状態のもの個数を保持することになる。プロセッサ数がPの場合、カウンタは0～P+ α までの値を取りうる。ここで、 α は正の整数であり、最近の高速プロセッサが有するライトバッファの効果により、システムバスを監視しているチェックポイント処理加速装置からは、一時的にはあるが、ひとつのキャッシュブロックに対応して、複数の番地のデータがModified状態で保持されているように見えることに対応するものである。 α の値は、P程度であれば十分である。なお、ブロックカウンタを構成する各カウンタは、ブロックカウンタ制御部37により増減される。また、キャッシュフラッシュ実行部34によって参照される。

【0183】ブロックカウンタ制御部37は、システムバス40上のコマンドおよびその応答を監視して、キャッシュブロックがModified状態に変わるのを検知したときに、そのアドレスに対応する前記カウンタがある場合は、それをインクリメントする。一方、キャッシュブロックがModified状態からそれ以外の状態に変わるのを検知したときには、そのアドレスに対応する前記カウンタがある場合は、それをデクリメントする。

【0184】ブロックカウンタ制御部37がシステムバス40上のコマンドおよびその応答を検知したときの動作を下記に示す。

【0185】(1) Read-Line

もし、モディファイド応答信号線がアサートされれば、コマンドに含まれるアドレスに対応するカウンタをデク

リメントする。そうでなければ何もしない。

【0186】(2) Read-Line-with-Invalidate

もし、モディファイド応答信号線がアサートされれば何もしない。そうでなければ、コマンドに含まれる更新アドレスに対応するカウンタをインクリメントする。

【0187】(3) Invalidate

コマンドに含まれる更新アドレスに対応するカウンタをインクリメントする。

【0188】(4) Write-Line

コマンドに含まれるアドレスに対応するカウンタをデクリメントする。

【0189】なお、Read-Lineコマンドでモディファイド応答信号線がアサートされた場合は、Modified状態のキャッシュブロックのデータがメインメモリに書き戻され、状態がModifiedからClean-Sharedに変化したことを示すから、カウンタをデクリメントする。また、Read-Line-with-Invalidateコマンドでモディファイド応答信号線がアサートされた場合は、Modified状態のキャッシュブロックがあるキャッシュメモリから他のキャッシュメモリに移動したということであるから、カウンタの値は変えない。

【0190】ビフォアイメージ取得部32は、第1の実施形態のビフォアイメージ取得部32と同一である。

【0191】キャッシュフラッシュ実行部34は、第1の実施形態のキャッシュフラッシュ実行部34に、Read-Lineコマンドを発行する際、対応するカウンタの値を参照し、もし、0であれば、Read-Lineコマンドを発行しないようにする機構を追加したものである。図17には、キャッシュフラッシュ実行部34の処理フローが示されている。

【0192】メインメモリ状態復元部33は、第1の実施形態と同じである。

【0193】ここで、チェックポイント処理加速装置30の動作を説明する。

【0194】(初期化) プロセッサ10は、ブロックカウンタ38を構成するすべてのカウンタの値を0とする。これと同期して、プロセッサ10はソフトウェアを実行することにより、すべてのキャッシュブロックをModified状態以外の状態にする。

【0195】(通常データ処理時) プロセッサ10は、通常データ処理を開始する時点で、ビフォアイメージ取得部32を起動する。ここでは、図18を参照し、通常データ処理として、プロセッサ10が、a番地のデータをA0からA1に、b番地のデータをB0からB1に、a番地のデータをA1からA2に、この順で更新する場合について説明する。ただし、a番地、b番地は、ともにキャッシュブロックBLK0に対応し、BLK0は、初期状態ではa番地のデータA0をClean

n-Shared状態で保持しているとする。また、キャッシュブロックBLK0に対応して、カウンタが1つ設けられているとする。

【0196】(1) 初期状態を示す。

【0197】(2) プロセッサ10がa番地のデータをA1に更新しようとするが、キャッシュブロックBLK0がa番地のデータをClean-Shared状態で保持しているため、a番地に対するInvalidateコマンドが発行される。

【0198】(3) そのデータを有する他のキャッシュメモリ20は、a番地のキャッシュブロックがあれば無効化し、終了したことを示す応答を返す。キャッシュメモリ20上のデータが、A0からA1へと更新される。ブロックカウンタ制御部37は、カウンタをインクリメントして1とする。

【0199】(4) ビフォアイメージ取得部32は、a番地に対してRead-Line-Non-Coherentコマンドを発行する。

【0200】(5) メインメモリ51がデータA0を応答する。ビフォアイメージ取得部32は、a番地とデータA0の組を、ビフォアイメージ記憶部35に格納する。

【0201】(6) プロセッサ10がb番地のデータをB1に更新しようとするが、キャッシュブロックBLK0がa番地のデータA1をModified状態で保持しているため、a番地のデータ書き戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にb番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0202】(7) メインメモリ51がデータB0を含む応答を返す。ブロックカウンタ制御部37は、カウンタをインクリメントして2とする。ビフォアイメージ取得部32は、b番地とデータB0の組をビフォアイメージ記憶部35に格納する。

【0203】(8) ライトバッファに保持されていたa番地に対するWrite-Lineコマンドが発行される。

【0204】(9) メインメモリ51のa番地のデータがA1に変わる。ブロックカウンタ制御部37は、カウンタをデクリメントして1とする。

【0205】(10) プロセッサ10がa番地のデータをA2に更新しようとするが、キャッシュブロックBLK0がb番地のデータB1をModified状態で保持しているため、b番地のデータ書き戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にa番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0206】(11) メインメモリ51がデータA1を含む応答を返す。ブロックカウンタ制御部37は、カウンタをインクリメントして2とする。ビフォアイメージ

取得部32は、a番地とデータA1の組をビフォアイメージ記憶部35に格納する。

【0207】(12) ライトバッファに保持されていたb番地に対するWrite-Lineコマンドが発行される。

【0208】(13) メインメモリ51のb番地のデータがB1に変わる。このとき、ブロックカウンタ制御部37は、カウンタをデクリメントして1とする。

【0209】以上、プロセッサ10、メインメモリ51、ブロックカウンタ制御部37、およびビフォアイメージ取得部32の動作例を示した。

【0210】(チェックポイント処理時) 前述した状態で、プロセッサ10がキャッシュフラッシュ実行部34を起動すると、キャッシュフラッシュ実行部34は、a番地、b番地、a番地の順にRead-Lineコマンドを発行する。その様子を図19に示す。ただし、ビフォアイメージの欄の下線は、Fポインタの指す位置を表わす。

【0211】(1) 初期状態を示す。

【0212】(2) キャッシュフラッシュ実行部34は、最初に、取得したビフォアイメージa番地とデータA0の組を使う。a番地に対応するカウンタが1であるので、a番地に対してRead-Lineコマンドを発行する。

【0213】(3) BLK0はa番地のデータA2をModified状態で保持しているため、モディファイド応答信号線をアサートし、メインメモリ51への書き戻しが行なわれ、メインメモリ51のa番地がA2に変わる。BLK0の状態は、Clean-Sharedに変化する。ブロックカウンタ制御部37は、Read-Lineコマンドに対して、モディファイド応答信号線がアサートされたので、カウンタをデクリメントして0とする。

【0214】(4) キャッシュフラッシュ実行部34は、ビフォアイメージb番地とデータB0の組を使おうとするが、b番地に対応するカウンタは0であるので、Read-Lineコマンドは発行しない。

【0215】(5) キャッシュフラッシュ実行部34は、ビフォアイメージa番地とデータA1の組を使おうとするが、a番地に対応するカウンタは0であるので、Read-Lineコマンドは発行しない。

【0216】(6) キャッシュフラッシュ実行部34の動作が終了すると、Modified状態であったすべてのキャッシュブロックの内容がメインメモリ51に書き戻されるとともに、それらキャッシュブロックの状態がClean-Shared状態になる。また、すべてのカウンタの値は0となる。これは、初期化直後と同じ状態であり、したがって、キャッシュフラッシュ実行部34の動作が終了すると、プロセッサ10は直ちに通常のデータ処理を始めることができる。

【0217】以上、ビフォアイメージを使った、キャッシュフラッシュ実行部34の動作例を示した。

【0218】このように、本実施形態では、ブロックカウンタ38を設けることにより、キャッシュフラッシュ実行部34が発行するRead-Lineコマンドの数を減らすことができるため、その分、チェックポイント処理の所要時間を短縮できる。

【0219】(障害発生時の復旧処理)次に、先に説明したチェックポイント処理の開始前に故障が発生した場合を例に図20を参照して説明する。障害が発生したとき、プロセッサ10はすべてのキャッシュブロックを無効化するので、キャッシュブロックBLK0の状態もInvalidとなっている。

【0220】(1)初期状態を示す。

【0221】(2)プロセッサ10がすべてのキャッシュブロックの無効化を行なう命令を実行する。

【0222】(3)この命令実行の結果、BLK0はInvalid状態となる。

【0223】(4)メインメモリ状態復元部33が、最後に取得したビフォアイメージa番地とデータA1の組をもとに、Write-Lineコマンド発行する。

【0224】(5)メインメモリ51のa番地にA1が書かれるが、もともとA1であったので変わらない。

【0225】(6)メインメモリ状態復元部33が、その前に取得したビフォアイメージb番地とデータB0の組をもとに、Write-Lineコマンド発行する。

【0226】(7)メインメモリ51のb番地がB0に変わる。

【0227】(8)メインメモリ状態復元部33が、その前に取得したビフォアイメージa番地とデータA0の組をもとに、Write-Lineコマンド発行する。

【0228】(9)メインメモリ51のa番地がA0に変わる。

【0229】(10)プロセッサ10が、すべてのカウンタの値を0にする。

【0230】この結果、メインメモリ51は、チェックポイント処理終了の時点の状態に戻る。

【0231】以上、メインメモリ状態復元部33の動作例を示した。

【0232】なお、ブロックカウンタ38は、すべてのアドレスに対して設定する必要は必ずしもない。あるアドレスについて、対応するカウンタがない場合、キャッシュフラッシュ実行部34は、そのアドレスに対するRead-Lineコマンドを常に行発するようにすれば良い。

【0233】このように、本実施形態のチェックポイント処理加速装置30では、ブロックカウンタ38を設けることにより、キャッシュフラッシュ実行部34がModified状態のキャッシュブロックが実在しないアドレスに対して、Read-Lineコマンドを発行す

るという無駄をある程度軽減できる。また、キャッシュフラッシュ実行部34が実行中も、ブロックカウンタ制御部37がブロックカウンタ38のインクリメント/デクリメントを適切に行なうことにより、キャッシュフラッシュ実行部34が実行を終了した時、すべてのブロックカウンタの値は0となっている。このため、キャッシュフラッシュ実行部34が実行を終了すると、プロセッサは直ちに通常のデータ処理を開始できる。

【0234】なお、前述した実施形態では、ブロックカウンタ38を、ダイレクトマップ方式のキャッシュメモリ20についてキャッシュブロックの各々に対応させて設ける方式を示した。

【0235】ここでは、キャッシュメモリ20がnウェイのセットアソシアティブ方式の場合について示す。

【0236】nウェイのセットアソシアティブ方式のキャッシュメモリの場合、あるアドレスのデータは、ひとつのウェイを形成するn個のキャッシュブロックのいずれかに格納されるが、そのうちのどのキャッシュブロックに格納されるかは、その都度変わらう。

【0237】そこで、ひとつのウェイを形成するn個のキャッシュブロックに対して、ひとつのカウンタを設ける。すなわち、あるカウンタは、対応するウェイでModified状態のキャッシュブロックの個数を保持することになる。プロセッサ数がPの場合、カウンタは0~ $P \times n + \alpha$ までの値を取りうる。 α はダイレクトマップ方式の場合に説明したように、プロセッサのライトバッファに対応するためのマージンである。

【0238】また、ブロックカウンタ38をキャッシュブロックあるいはひとつのウェイを形成するキャッシュブロックの集合に対応させて設けることに代えて、メインメモリをキャッシュブロックの大きさに分け、それぞれに対してカウンタを設けることも可能である。この場合、計算機のプロセッサ数に関係なく、ひとつのアドレスに対するModified状態のキャッシュブロックの個数は0か1かであるので、それぞれ1ビットで表現することができ、ブロックカウンタ制御部37を簡単化することができる。

【0239】これは、メインメモリをキャッシュブロックの大きさに分けて、1ビットの情報を持たせるという点で、第2の実施形態と類似するが、下記の点で異なる。

【0240】すなわち、1ビット情報のオン/オフするタイミング、および、そのオン/オフによって制御する対象が異なる。具体的には、第2の実施形態では、ビフォアイメージを取得するか否かの判断に使用するのに対し、キャッシュフラッシュ実行部34がRead-Lineコマンドを発行するか否かの判断に使用する。

【0241】また、チェックポイント方式の計算機の障害が発生した場合、すべてのメインメモリを直前のチェックポイントに戻すのではなく、メインメモリの一部の

領域、たとえば障害発生と回復処理の履歴を保持する領域や、回復処理を実行するプログラムの作業領域など、戻してはならない領域が存在する。

【0242】このチェックポイント処理加速装置30で、それを実現する方式には下記の2種類の候補がある。

【0243】(1) ビフォアイメージ取得部32に、ビフォアイメージを取得する可否かを判断させる機構を設ける。この方法は、取得するビフォアイメージが少なくなる長所があるが、その判断を高速に実行しないと、プロセッサ10やシステムバス40の動作に追従できなくなる危険性がある。

【0244】(2) メインメモリ状態復元部33に、同様の判断機構を持たせ、メインメモリ51にビフォアイメージを書き込むWrite-Lineコマンドを発行する可否かを判断する。この方法では、実行速度を余り気にする必要はない。また、同様の判断機構を、キャッシュフラッシュ実行部34に持たせれば、チェックポイント処理の所要時間が若干短縮できる。

【0245】ここでは、状態を復元してはならない領域は、制御ソフトウェアの方で、物理的に連続な領域で、かつ、領域の先頭アドレスをある程度自由に設定できるという前提で、前者にも適用可能な判断が高速でハードウェア量も余り必要としない方式を示す。

【0246】図21のように、この機構は、判断の対象となる更新アドレスを保持するアドレスレジスタ、アドレスレジスタの一部ビットをマスクするためのマスクレジスタと32ビットのAND回路、その結果と比較するための比較参照レジスタと32ビット比較器で構成される。そして、比較器で一致と判断したときは、ビフォアイメージを採取しないようにする。なお、この機構でも判断が遅すぎる場合には、たとえばInvalidateコマンドに対するRead-Line-Non-Coherentコマンドを先に発行して、ビフォアイメージを格納する直前にこの機構の結果を使用しても良い。

【0247】これまで、キャッシュメモリ20上でデータ更新が発生した場合に、システムバス40に発行されるInvalidateコマンド、およびRead-Line-with-Invalidateコマンドが観測されたとき、ビフォアイメージ取得部32が更新アドレスと更新前データを取得する方法を示してきた。しかしながら、計算機によっては、I/O機器からメインメモリ51へのデータ転送や、プロセッサ10からのキャッシュメモリ20を通さないメインメモリ51への書き込みが起こる場合があるので、次に、その対策について説明する。

【0248】(対策1) ここでは、下記の前提に基づいて、第1の対策について説明する。

【0249】(1) メインメモリ51への書き込みには、Write-Lineコマンドとは別のコマンド

(ここでは、それをWrite-Non-Coherentコマンドと呼ぶ) が使われる。

【0250】(2) システムバス40の仕様として、Retry応答線という制御信号線があり、Write-Non-Coherentコマンドがシステムバスに発行されたとき、Retry信号線をアサートすることにより、そのコマンドの実行中止を要求できる。この場合、暫くして、中止させられたコマンドが再度発行される。

【0251】ビフォアイメージ取得部32は、Write-Non-Coherentコマンドを検出すると、Retry応答線をアサートする。そして、その直後に、Write-Non-Coherentコマンドに含まれている更新アドレスを使って、Read-Line-Non-Coherentコマンドを発行する。そして、メインメモリ51からの応答に含まれるデータと前記更新アドレスの組をビフォアイメージとしてビフォアイメージ記憶部35に格納する。一方、ビフォアイメージ取得部32は、先程と同じアドレスを含むWrite-Non-Coherentコマンドを検出したときには、Retry応答線をアサートしない。

【0252】(対策2) I/O機器からメインメモリ51へのデータ転送や、プロセッサ10からのキャッシュメモリ20を通さないメインメモリ51への書き込みは、プロセッサ10の制御下で行なわれる。したがって、Write-Non-Coherentコマンドが発行される前に、プロセッサ10がそのアドレスに対してキャッシュメモリ20を経由したデータ更新を行なうことにより、ビフォアイメージ取得部32にビフォアイメージを取得させる。次に、プロセッサ10は、更新したデータを保持しているキャッシュブロックを無効化し、その後、I/O機器からメインメモリ51へのデータ転送などを起動する。

【0253】(対策3) ビフォアイメージ取得部32は、プロセッサ10から指定されたアドレス範囲について、ビフォアイメージの取得と格納を実行する機能を付加する。通常、I/O機器からメインメモリ51へのデータ転送は、連続したアドレスに対して行なわれるので、プロセッサ10からの1回の指示で複数のビフォアイメージの取得が可能となる。したがって、対策2より処理が高速である。

【0254】以上、第3の実施形態について説明した。

【0255】(第4の実施形態) 次に、この発明の第4の実施形態を説明する。前述した第1乃至第3の実施形態では、通常の前データ処理時、ビフォアイメージ取得部32を動作させることによってビフォアイメージを取得し、チェックポイント処理時には、プロセッサ10がキャッシュフラッシュ実行部34を起動し、その終了を待って通常の前データ処理を再開する制御方式のもとでの動作説明をした。ここでは、チェックポイント処理の所要

時間を短縮する、より最適化されたチェックポイント処理加速装置 3 0 の制御方法、およびその制御方法に適合したチェックポイント処理加速装置の機能および構成法を示す。なお、この制御方式をフラッシュ前倒し方式と呼ぶことにする。

【0256】このチェックポイント処理加速装置 3 0 を適用したチェックポイント／リカバリ方式の計算機では、通常の前データ処理を継続する時間は、典型的には数ミリ秒～数 1 0 ミリ秒である。チェックポイント処理が終了した時点では、すべてのキャッシュブロックが、Invalid 状態、または Clean-Shared 状態であり、Modified 状態のものはない。そして、通常の前データ処理を再開すると、次第に Modified 状態のキャッシュブロックが増加し、次のチェックポイント処理を始める時には、典型的には、1 0 %～5 0 % のキャッシュブロックが Modified 状態になっている。

【0257】チェックポイント処理の所要時間の大部分は、ビフォアイメージ記憶部 3 5 に格納されているビフォアイメージのアドレスに対して、Read-Line コマンドを発行する処理が占める。これを高速化する方法として、第 2 の実施形態ではフラグメモリ 3 6 を使用する方法を、また、第 3 の実施形態ではブロックカウンタ 3 8 を使用する方法を示した。そして、これらは無駄な Read-Line コマンドの発行を減らすことを目的としていた。

【0258】ここでは、キャッシュフラッシュ実行部 3 4 の起動を、チェックポイント処理の開始前、すなわち、通常の前データ処理中に行なう方法を説明する。また、第 1 乃至第 3 の実施形態で説明したチェックポイント処理加速装置が、いずれもこの制御方式のもとで正しく動作すること、および、この制御方式により適合したチェックポイント処理加速装置の構成法について説明する。

【0259】最初に、図 2 2 に基づいて、プロセッサ 1 0、ビフォアイメージ取得部 3 2、およびキャッシュフラッシュ実行部 3 4 の動作の時間的な関係を説明する。

【0260】プロセッサ 1 0 は、通常の前データ処理およびビフォアイメージ処理の前半は、ビフォアイメージ取得部 3 2 を実行させておく。また、プロセッサ 1 0 は、通常の前データ処理の途中で、キャッシュフラッシュ実行部 3 4 を起動する。すると、その後は、プロセッサ 1 0 が通常の前データ処理にともなって、データの更新を行ない、それをビフォアイメージ取得部 3 2 がビフォアイメージ記憶部 3 5 の W ポインタ（第 1 の実施形態で説明）が指すエントリに格納するという処理と、キャッシュフラッシュ実行部 3 4 がビフォアイメージ記憶部 3 5 の F ポインタが指すエントリに格納されているアドレスに対して Read-Line コマンドを発行することにより、Modified 状態のキャッシュブロックの内容

をメインメモリに書き戻すという処理が並行して行なわれる。

【0261】そして、プロセッサ 1 0 がチェックポイント処理を開始すると、まず、通常の前データ処理を中断した時のコンテキストのメインメモリ 5 1（実際には、キャッシュメモリ 2 0）への書き込み後、キャッシュフラッシュ実行部 3 4 が処理を終了するのを待つ。

【0262】次に、フラッシュ前倒し方式のプロセッサの処理フローを図 2 3 に基づいて説明する。

【0263】通常の前データ処理では、ビフォアイメージ取得部 3 2 を起動した後、アプリケーションプログラムやオペレーティングシステムを実行する。そして、適当なタイミングになるとキャッシュフラッシュ実行部 3 4 を起動する。このキャッシュフラッシュ実行部 3 4 を起動するタイミングについては後述する。そして、さらにアプリケーションプログラムやオペレーティングシステムの実行を続け、適当なタイミングになると、通常の前データ処理を中断し、チェックポイント処理を開始する。

【0264】チェックポイント処理では、まず、プロセッサ 1 0 を割り込み禁止状態にして、外部機器などからの割り込みを受けつけないようにする。これは、チェックポイント処理中は、通常の前データ処理を一切実行しないようにするためである。ただし、計算機の故障発生を通知する割り込みについては、通常の前データ処理により高い優先度を設定するなど、割り込み禁止状態にあっても受け付けられるようになっている必要がある。

【0265】次に、プロセッサ 1 0 は、通常の前データ処理を中断した時点でのコンテキスト、すなわちレジスタの内容のメインメモリ 5 1 の然るべきアドレスへの書き込みを行なう。ただし、本実施形態では、コピーバック型のキャッシュメモリが使用されるので、この時点では通常の前データ更新と同じように、キャッシュブロックに Modified 状態で保持され、ビフォアイメージ記憶部 3 5 にそのアドレスのビフォアイメージが記憶されている。

【0266】その後、プロセッサ 1 0 は、キャッシュフラッシュ実行部 3 4 がビフォアイメージ記憶部 3 5 内のビフォアイメージを使って、すべての Modified 状態のキャッシュブロックのデータがメインメモリ 5 1 に書き戻されるのを待つ。プロセッサ 1 0 のコンテキストも、このとき、メインメモリ 5 1 に書き戻される。キャッシュフラッシュ実行部 3 4 が実行を終了すると、メインメモリ 5 1 に保存したコンテキストを対応するレジスタに戻し、割り込み禁止状態を解除することにより、通常の前データ処理を再開する。

【0267】フラッシュ前倒し方式の効果を、元の制御方式と対比させて模式的に表わすと、図 2 4 のようになる。

【0268】元の制御方式（a）もフラッシュ前倒し方式（b）も、通常の前データ処理を開始した直後は、新た

に取得されるビフォアイメージの量が、ほぼ直線的に増加する。

【0269】元の制御方式では、この状態がチェックポイント処理の直前まで続くのに対して、フラッシュ前倒し方式では、通常のデータ処理の途中でキャッシュフラッシュ実行部34を起動することにより、ビフォアイメージ取得部32が新たにビフォアイメージを取得する一方でキャッシュフラッシュ実行部34がRead-Lineコマンドを発行し、その結果、キャッシュフラッシュ実行部34が処理すべきビフォアイメージの量(WポイントとFポイントの差)の増加が鈍るもしくは減る。模式図では減る場合を示した。キャッシュフラッシュ実行部34が処理すべきビフォアイメージの量を減らすようにするか否かは、キャッシュフラッシュ実行部34の構成法で制御できるので、後で説明する。

【0270】このチェックポイント処理加速装置30を用いた場合のチェックポイント処理の所要時間は、ほぼ、キャッシュフラッシュ実行部34が処理すべきビフォアイメージの量に比例するので、フラッシュ前倒し方式は、チェックポイント処理の所要時間を大幅に減らすことができる。通常のデータ処理の継続時間は、前に述べた通り、典型的には数ミリ秒～数10ミリ秒の間であり、元の制御方式の場合、チェックポイント処理の所要時間は、1ミリ秒～10ミリ秒程度である。その間、外部機器からの割込みを受け付けないことが、本チェックポイント／リカバリ方式の計算機の適用範囲を制限する要因となりうる。したがって、フラッシュ前倒し方式により、チェックポイント処理の所要時間が、たとえば数分の1になるということは十分に意味がある。

【0271】ただし、フラッシュ前倒し方式には、性能を劣化させる側面もある。それは、将来もModified状態で使われるキャッシュブロック(メインメモリ上に設けられたカウンタがその典型である)をClean-Shared状態にしてしまい、次にプロセッサがそのデータを更新しようとするとき、Invalidateコマンドが発行されてしまう点である。この性能劣化を軽減する方法については、後で説明する。

【0272】ここでチェックポイント処理加速装置30の動作を説明する。

【0273】フラッシュ前倒し方式のもとで、第1乃至第3の実施形態のチェックポイント処理加速装置30は正しく動作する。動作例は類似するので、ここでは、第2の実施形態の場合について説明するにとどめる。

【0274】(初期化)プロセッサ10は、すべてのフラグメモリ36をオフにする。これと同期して、プロセッサ10はソフトウェアを実行することにより、すべてのキャッシュブロックをModified状態以外の状態にする。

【0275】(通常のデータ処理およびチェックポイント処理時)プロセッサ10は、通常のデータ処理を開始

する時点で、ビフォアイメージ取得部32を起動する。ここでは、下記に示した順序で処理が行なわれた場合について説明する。

【0276】(1)プロセッサ10が、a番地のデータをA0からA1に、b番地のデータをB0からB1に、この順で更新した。

【0277】(2)プロセッサ10が、キャッシュフラッシュ実行部34を起動した。

【0278】(3)キャッシュフラッシュ実行部34が、a番地に対してRead-Lineコマンドを発行した。

【0279】(4)プロセッサ10が、a番地のデータをA1からA2に更新した。

【0280】ただし、a番地、b番地、ともにキャッシュブロックBLK0に対応し、BLK0は、初期状態ではa番地のデータA0をClean-Shared状態で保持しているとする。

【0281】この場合の動作を図25を参照して説明する。

【0282】(1)初期状態を示す。

【0283】(2)プロセッサ10がa番地のデータをA1に更新しようとするが、キャッシュブロックBLK0がa番地のデータをClean-Shared状態で保持しているため、a番地に対するInvalidateコマンドが発行される。

【0284】(3)そのデータを有する他のキャッシュメモリ20は、a番地のキャッシュブロックがあれば無効化し、終了したことを示す応答を返す。

【0285】(4)ビフォアイメージ取得部32は、a番地に対応するフラグメモリ36がオフであるため、これをオンにし、a番地に対してRead-Line-Non-Coherentコマンドを発行する。

【0286】(5)メインメモリ51がデータA0を応答する。ビフォアイメージ取得部32が、a番地とデータA0の組をビフォアイメージ記憶部35に格納する。

【0287】(6)プロセッサ10がb番地のデータをB1に更新しようとするが、キャッシュブロックBLK0がa番地のデータA1をModified状態で保持しているため、a番地のデータ書き戻しのためのWrite-Lineコマンドがライトバッファに一時的に保持され、先にb番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0288】(7)メインメモリ51がデータB0を応答する。ビフォアイメージ取得部32は、b番地に対応するフラグメモリ36がオフであるため、これをオンにし、b番地とデータB0の組をビフォアイメージ記憶部35に格納する。

【0289】(8)ライトバッファに保持されていたa番地に対するWrite-Lineコマンドが発行される。

【0290】(9) メインメモリのa番地がA1に変わる。

【0291】(10) プロセッサ10は、この時点でキャッシュフラッシュ実行部34を起動する。

【0292】(11) キャッシュフラッシュ実行部34は、ビフォアイメージa番地とデータA0の組を使い、a番地に対応するフラグメモリ36をオフにするとともに、a番地に対してRead-Lineコマンドを発行する。

【0293】(12) BLK0はb番地のデータB1をModified状態で保持しているので、モディファイド応答信号線もシェアード応答信号線もアサートされない。メインメモリ51がA1を応答する。BLK0の状態は変化しない。

【0294】(13) キャッシュフラッシュ実行部34は、ビフォアイメージb番地とデータB0の組を使い、b番地に対応するフラグメモリ36をオフにするとともに、b番地に対してRead-Lineコマンドを発行する。

【0295】(14) BLK0はb番地のデータB1をModified状態で保持しているので、モディファイド応答信号線がアサートされ、メインメモリ51への書き戻しが行なわれ、メインメモリ51のb番地がB1に変わる。BLK0の状態は、Clean-Sharedに変わる。

【0296】(15) プロセッサ10がa番地のデータをA2に更新しようとするが、キャッシュブロックBLK0がb番地のデータB1をClean-Shared状態で保持しているため、a番地に対するRead-Line-with-Invalidateコマンドが発行される。

【0297】(16) メインメモリ51がデータA1を含む応答を返す。ビフォアイメージ取得部32は、a番地に対応するフラグメモリ36がオフであるため、これをオンにし、a番地とデータA1の組をビフォアイメージ記憶部35に格納する。

【0298】(17) キャッシュフラッシュ実行部34は、ビフォアイメージa番地とデータA1の組を使い、a番地に対応するフラグメモリ36をオフにし、a番地に対してRead-Lineコマンドを発行する。

【0299】(18) BLK0はa番地のデータA2をModified状態で保持しているので、モディファイド応答信号線がアサートされ、メインメモリ51への書き戻しが行なわれ、メインメモリ51のa番地はA2に変わる。BLK0の状態は、Clean-Sharedに変わる。

【0300】以上、本制御方法における、プロセッサ10、メインメモリ51、ビフォアイメージ取得部32、およびキャッシュフラッシュ実行部34の動作例を示した。

【0301】(障害発生時の復旧処理) メインメモリ状態復元部33の動作は、ビフォアイメージ記憶部35に記憶されているビフォアイメージを、より後で格納されたものから順に、Write-Lineコマンドを発行するだけであるので、動作例に基づいた説明を割愛する。

【0302】以上、フラッシュ前倒し方式のもとで、本発明の第2の実施形態のチェックポイント処理加速装置が正しく動作することを説明した。

【0303】次に、チェックポイント処理加速装置30の機能および構成法を説明する。

【0304】第1乃至第3の実施形態では、キャッシュフラッシュ実行部34は、ビフォアイメージ記憶部35に格納されているビフォアイメージをFポインタを使って最初に格納されたものから順に処理する方式を示した。元の制御方式の場合には、ビフォアイメージをどのような順序で処理しても大して変わりはないが、フラッシュ前倒し方式の場合、この処理方式は2つの長所がある。

【0305】第1の長所は、将来もModified状態で使われるキャッシュブロックをClean-Shared状態にしてしまい、次にプロセッサがそのデータを更新しようとする時、Invalidateコマンドが発行されてしまうという性能劣化要因を軽減できる点である。

【0306】つまり、早い時点で取得したビフォアイメージほど、キャッシュフラッシュ実行部34が処理しようとする時点では、すでにModified状態のキャッシュブロックが実在しない可能性が高く、キャッシュフラッシュ実行部34がRead-Lineコマンドを発行しても、それによってClean-Shared状態に変化する可能性は小さいためである。

【0307】第2の長所は、ハードウェアの制御のしやすさである。まず、ビフォアイメージ取得部32がWポインタを使って最初のエントリから順に書き込むのを、キャッシュフラッシュ実行部34がFポインタを使って追いかけるため、キャッシュフラッシュ実行部34の終了の判断が簡単になる。

【0308】また、フラッシュ前倒し方式では、ビフォアイメージ取得部32とキャッシュフラッシュ実行部34によるビフォアイメージ記憶部35へのアクセス競合が発生する。ひとつのメモリバンクに対してリードアクセスとライトアクセスが交互に出ると、リードアクセスが遅くなるので、そういった状況避けることが望ましい。特に、複数プロセッサを備えたマルチプロセッサシステムに適用する場合には、アクセス競合の回避が重要である。

【0309】ここでは、第1の実施形態で述べたWポインタとFポインタを使用することを前提に、アクセス競合の少ないビフォアイメージ記憶部35のエントリの構

成法とその使用法を示す。

【0310】まず、前半のn個のエントリと、後半のn個のエントリを、図26に示したように、メモリバンクAとメモリバンクBとに分けて構成する。ただし、2つのバンクメモリに分けるのは、それぞれのバンクメモリに同時にアクセス可能とすることと、後で示すように、ひとつのメモリバンクに対してリードアクセスとライトアクセスが交互に出るような状況を作らないためである。

【0311】通常のデータ処理の前半、すなわち、ビフォアイメージ取得部32が動作し、キャッシュフラッシュ実行部34が動作しないときは、取得したビフォアイメージをメモリバンクAに格納する。このとき、メモリバンクAは、ライトアクセスのみとなる。

【0312】次に、プロセッサ10がキャッシュフラッシュ実行部34を起動すると、ビフォアイメージ取得部32は、取得したビフォアイメージをメモリバンクBに格納する。そして、キャッシュフラッシュ実行部34は、メモリバンクAからビフォアイメージを取り出す。このとき、メモリバンクAはリードアクセスのみ、メモリバンクBはライトアクセスのみとなる。

【0313】次に、プロセッサ10は、キャッシュフラッシュがメモリバンクA内のビフォアイメージを全部処理する直前にチェックポイント処理を開始し、間もなくビフォアイメージ取得部32を停止する。このとき、メモリバンクBは、ほぼライトアクセスのみとなる。

【0314】以上のように、メモリバンクの使用状況とキャッシュフラッシュ実行部34の起動、およびチェックポイント処理の開始をリンクすることにより、各メモリバンクは、ほとんど常時、リードかライトのいずれかのモードでアクセスされることになり、ビフォアイメージの格納と取り出しとが高速化できる。

【0315】なお、前述の説明では、キャッシュフラッシュ実行部34はプロセッサ10が起動するとしたが、このように制御するのであれば、ビフォアイメージ取得部32が、メモリバンクAのすべてのエントリにビフォアイメージを格納した時点でキャッシュフラッシュ実行部34を起動するようにもでき、その場合、プロセッサ10は、キャッシュフラッシュ実行部34を起動する処理を省けるほか、メモリバンクAを容量一杯まで使用できるという長所がある。

【0316】また、キャッシュフラッシュ実行部34は、Read-Lineコマンドの発行頻度を制御することができる。元の制御方式のもとでは、キャッシュフラッシュ実行部34は、プロセッサ10がほとんど動作を停止している状況で動作するので、ビフォアイメージ記憶部35に格納されている更新アドレスを使って、いかに高速にRead-Lineコマンドを発行するかが重要であった。

【0317】しかし、フラッシュ前倒し方式のもとで

は、データ処理の途中において、キャッシュフラッシュ実行部34によるRead-Lineコマンドの発行頻度が過度に高いと、通常のデータ処理がほとんど進まないという状況となる。

【0318】そこで、キャッシュフラッシュ実行部34は、Read-Lineコマンドの発行頻度を少なくするモードと高いモードとを備えるのが望ましい。その実現方法の例を図27に示すキャッシュフラッシュ実行部34の処理フローに基づいて説明する。

【0319】この処理フローは、第1の実施形態を若干修正したものである。すなわち、Read-Lineコマンドを発行し、Fポインタをインクリメントしたところで、ビフォアイメージ取得部32が実行中であればCサイクル待ち、そうでなければ直ちに次のビフォアイメージを使ったRead-Lineコマンドを発行する。

【0320】この方式の変形としては、たとえばビフォアイメージ取得部32が最近の数10サイクルにおいて、ビフォアイメージを取得したことがある場合には、Cサイクル待つという方式も可能である。

【0321】以上、第4の実施形態について説明した。

【0322】(第5の実施形態)次に、この発明の第5の実施形態を説明する。前述した第4実施形態では、プロセッサ10は、チェックポイント作成時、キャッシュフラッシュ実行部34が終了するのを、ただ待つことに多くの時間を費やしていた。それを改善する方法として、チェックポイント作成時に、プロセッサがmodified状態のキャッシュブロックの内容をメインメモリ51に書き戻すキャッシュ命令を実行する方法について説明する。

【0323】図28は、本実施形態のプロセッサ10がチェックポイント加速装置30を使ってチェックポイント/ロールバック方式を実現する場合の通常のデータ処理およびチェックポイント処理における処理フローを示している。

【0324】通常のデータ処理は、第4の実施形態のフラッシュ前倒し方式と全く同一であるので説明は略する。

【0325】チェックポイント処理では、まず、プロセッサ10を割込み禁止状態にして、外部機器などからの割り込みを受け付けないようにする。

【0326】次に、プロセッサ10は、通常のデータ処理を中断した時点でのコンテキスト、すなわちレジスタの内容のメインメモリ51の然るべきアドレスへの書き込みを行なう。次に、プロセッサ10は、ひとつのキャッシュブロックのタグを読み出すキャッシュ命令を実行し、もし、modified状態であれば、そのキャッシュブロックの内容をメインメモリ51に書き戻すキャッシュ命令を実行するという処理をN回繰り返すことにより、N個のキャッシュブロックについてキャッシュフラッシュ処理を行なう。

【0327】そして、この時点で、キャッシュ命令によるキャッシュフラッシュ処理が終了していれば、キャッシュフラッシュ実行部34を停止させた後、メインメモリ51に保存したコンテキストを対応するレジスタに戻し、割り込み禁止状態を解除することにより、通常の実行部34のデータ処理を再開する。

【0328】そうでなければ、この処理の間、キャッシュフラッシュ実行部34も実行を継続しているので、プロセッサ10は、次にキャッシュフラッシュ実行部34が実行を終了したか調べる。もし、実行を終了していれば、メインメモリ51に保存したコンテキストを対応するレジスタに戻し、割り込み禁止状態を解除することにより、通常の実行部34のデータ処理を再開する。

【0329】この方式の効果は、チェックポイント処理の所要時間が短縮されることである。特に、キャッシュフラッシュ実行部34だけではシステムバスの転送能力に余裕が生ずる程にシステムバス40の転送能力が非常に大きい場合、キャッシュ命令によるキャッシュフラッシュを併用することで、チェックポイント処理の所要時間が短縮できる。

【0330】また、この方式の変形として、チェックポイント処理を始めた時点で、チェックポイント実行部34を途中で停止させて、キャッシュ命令によるキャッシュフラッシュ処理だけを行なう方法もある。もともと、キャッシュ命令によるキャッシュフラッシュ処理は、キャッシュフラッシュ実行部34と異なり、システムバス40に無用なバスコマンドを発行しないという利点がある。したがって、プロセッサ数がある程度多く、すべてのプロセッサが一斉にキャッシュ命令によるキャッシュフラッシュ処理したときに、システムバス40の転送能力を使いきるような場合には、本方式が有効である。

【0331】さらに、フラッシュ前倒し方式のもとで、チェックポイント処理をキャッシュ操作命令によるキャッシュフラッシュ処理だけを行なう方法（キャッシュ命令単独方式と呼ぶ）を前提として、チェックポイント処理加速装置を次のように改良できる。

【0332】第4の実施形態で説明したチェックポイント処理加速装置30では、キャッシュフラッシュ実行部34は、最初に取得したビフォアイメージから順に使用するものであった。しかし、キャッシュ命令単独方式の場合、チェックポイント処理加速装置30のキャッシュフラッシュ実行部34は、どのビフォアイメージを使用しても構わない。そこで、システムの性能を損なわないように、何度も繰り返し書き込みが発生するようなキャッシュブロックに対して、その内容をメインメモリへ書き戻すことのないようにするのが望ましい。

【0333】それをある程度実現する手段として、キャッシュフラッシュ実行部34は、最初に取得したビフォアイメージからではなく、途中で取得したビフォアイメージを使うようにすることが考えられる。というのは、

最初の方で取得したビフォアイメージに含まれるアドレスは、その後も頻繁に使われる可能性が、途中で取得したビフォアイメージに含まれるアドレスの場合より、高いことが多いと考えられるからである。

【0334】

【発明の効果】以上詳述したように、この発明のチェックポイント処理加速装置によれば、プロセッサがキャッシュメモリ上でのデータ更新を行なったときに、ビフォアイメージ取得部が、その更新アドレスと更新前データとを取得して、ビフォアイメージ記憶部に格納する。そして、キャッシュフラッシュ実行部は、プロセッサから指示されたときに、ビフォアイメージ記憶部に格納されているすべてのアドレスを使用して、そのアドレスを有する更新状態のキャッシュブロックの内容をメインメモリに書き戻すことと要求するコマンドをシステムバスに発行することによってキャッシュフラッシュを実行する。一方、メインメモリ状態復元部は、プロセッサから指示されたときに、ビフォアイメージ記憶部に格納されているすべての更新前データを新しく格納したものから順に、メインメモリの更新書き込みを要求するコマンドをシステムバスに発行することによってメインメモリをビフォアイメージ記憶部には何も記憶されていなかったときの状態に復旧する。

【0335】すなわち、このチェックポイント処理加速装置を適用すれば、特別なキャッシュメモリを備えるなどのことを必要とせずに、標準的なプロセッサを搭載する計算機システム上において効率的なチェックポイント／リカバリ機能を実現することが可能となる。また、従来のように、プロセッサがキャッシュフラッシュ用のソフトウェアを実行することによってキャッシュフラッシュを実施することと比較して、大幅な高速化が期待されるため、チェックポイント／リカバリ方式の計算機全体の性能を向上させることが可能となる。

【0336】また、キャッシュメモリ上に確保されるキャッシュブロックに対応させてフラグメモリを設け、ビフォアイメージの重複格納を回避する、あるいは、互いに共通部分をもたないようにアドレス範囲を設定し、この設定したアドレス範囲それぞれに対応させてブロックカウンタを設け、不要なキャッシュフラッシュ動作を回避するといったことを行なえば、さらなる性能向上を図れることになる。

【図面の簡単な説明】

【図1】この発明の第1の実施形態に係る計算機システムのシステム構成を示す図。

【図2】同実施形態のビフォアイメージ記憶部の構造を示す概念図。

【図3】同実施形態のビフォアイメージ取得部の処理フローを示す図。

【図4】同実施形態のキャッシュフラッシュ実行部の処理フローを示す図。

【図5】同実施形態のメインメモリ状態復元部の処理フローを示す図。

【図6】同実施形態のプロセッサによるチェックポイント処理加速装置の制御方式を示す図。

【図7】同実施形態のプロセッサによるデータ更新に伴う動作を説明する図。

【図8】同実施形態のキャッシュフラッシュ実行部によるRead-Lineコマンドの発行を説明する図。

【図9】同実施形態のチェックポイント処理の開始前に故障が発生した場合の動作を説明する図。

【図10】この発明の第2の実施形態に係る計算機システムのシステム構成を示す図。

【図11】同実施形態のビフォアイメージ取得部の処理フローを示す図。

【図12】同実施形態のキャッシュフラッシュ実行部の処理フローを示す図。

【図13】同実施形態のプロセッサによるデータ更新に伴う動作を説明する図。

【図14】同実施形態のキャッシュフラッシュ実行部によるRead-Lineコマンドの発行を説明する図。

【図15】同実施形態のチェックポイント処理の開始前に故障が発生した場合の動作を説明する図。

【図16】この発明の第3の実施形態に係る計算機システムのシステム構成を示す図。

【図17】同実施形態のキャッシュフラッシュ実行部の処理フローを示す図。

【図18】同実施形態のプロセッサによるデータ更新に伴う動作を説明する図。

【図19】同実施形態のキャッシュフラッシュ実行部によるRead-Lineコマンドの発行を説明する図。

【図20】同実施形態のチェックポイント処理の開始前に故障が発生した場合の動作を説明する図。

【図21】同実施形態のビフォアイメージを取得するか否かを判断する機構を示す図。

【図22】この発明の第4実施形態のプロセッサ、ビフォアイメージ取得部、およびキャッシュフラッシュ実行部の動作の時間的な関係を説明する図。

【図23】同実施形態のフラッシュ前倒し方式のプロセッサの処理フローを示す図。

【図24】同実施形態のフラッシュ前倒し方式の効果を元の制御方式と対比させて模式的に表わした図。

【図25】同実施形態のプロセッサ、メインメモリ、ビフォアイメージ取得部、およびキャッシュフラッシュ実行部の動作例を示す図。

【図26】同実施形態のビフォアイメージ記憶部のエントリの構成法とその使用法を説明する図。

【図27】同実施形態のキャッシュフラッシュ実行部の処理フローを示す図。

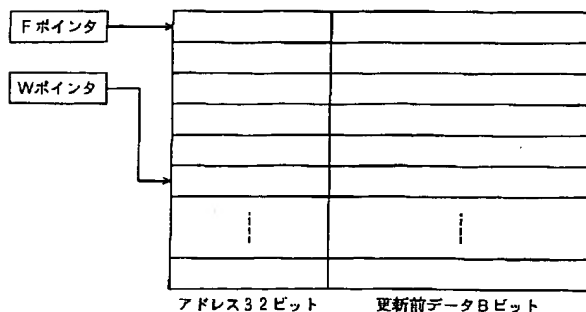
【図28】この発明の第5の実施形態のプロセッサがチェックポイント加速装置を使ってチェックポイント／ロールバック方式を実現する場合の通常データ処理およびチェックポイント処理における処理フローを示す図。

【図29】チェックポイント／リカバリ方式の動作手順を説明する図。

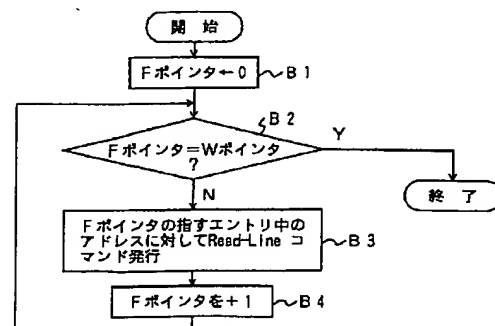
【符号の説明】

10…プロセッサ、20…キャッシュメモリ、30…チェックポイント処理加速装置、31…バスインタフェース、32…ビフォアイメージ取得部、33…メインメモリ状態復元部、34…キャッシュフラッシュ実行部、35…ビフォアイメージ記憶部、36…フラグメモリ、37…ブロックカウンタ制御部、38…ブロックカウンタ、40…システムバス、50…メモリコントローラ、51…メインメモリ。

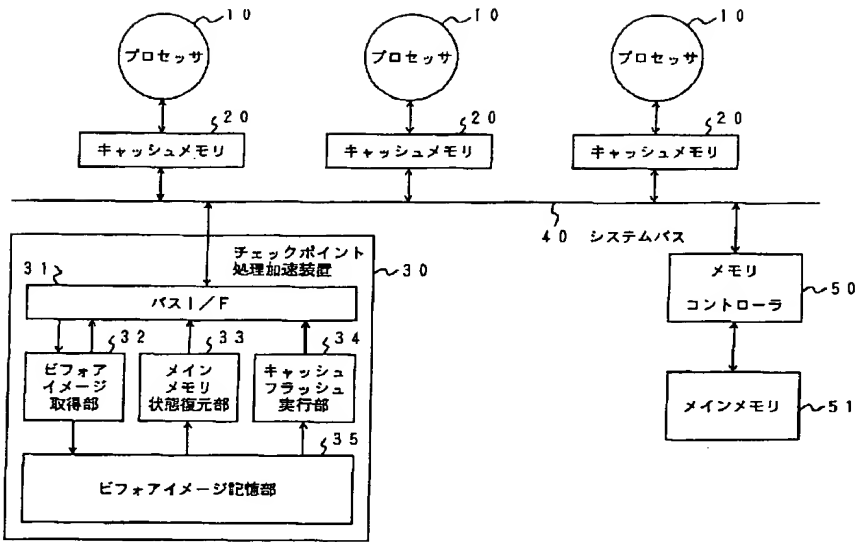
【図2】



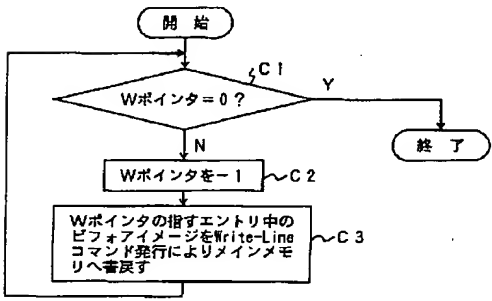
【図4】



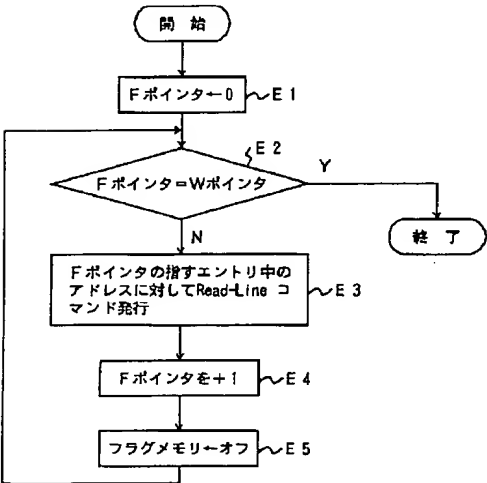
【図 1】



【図 5】



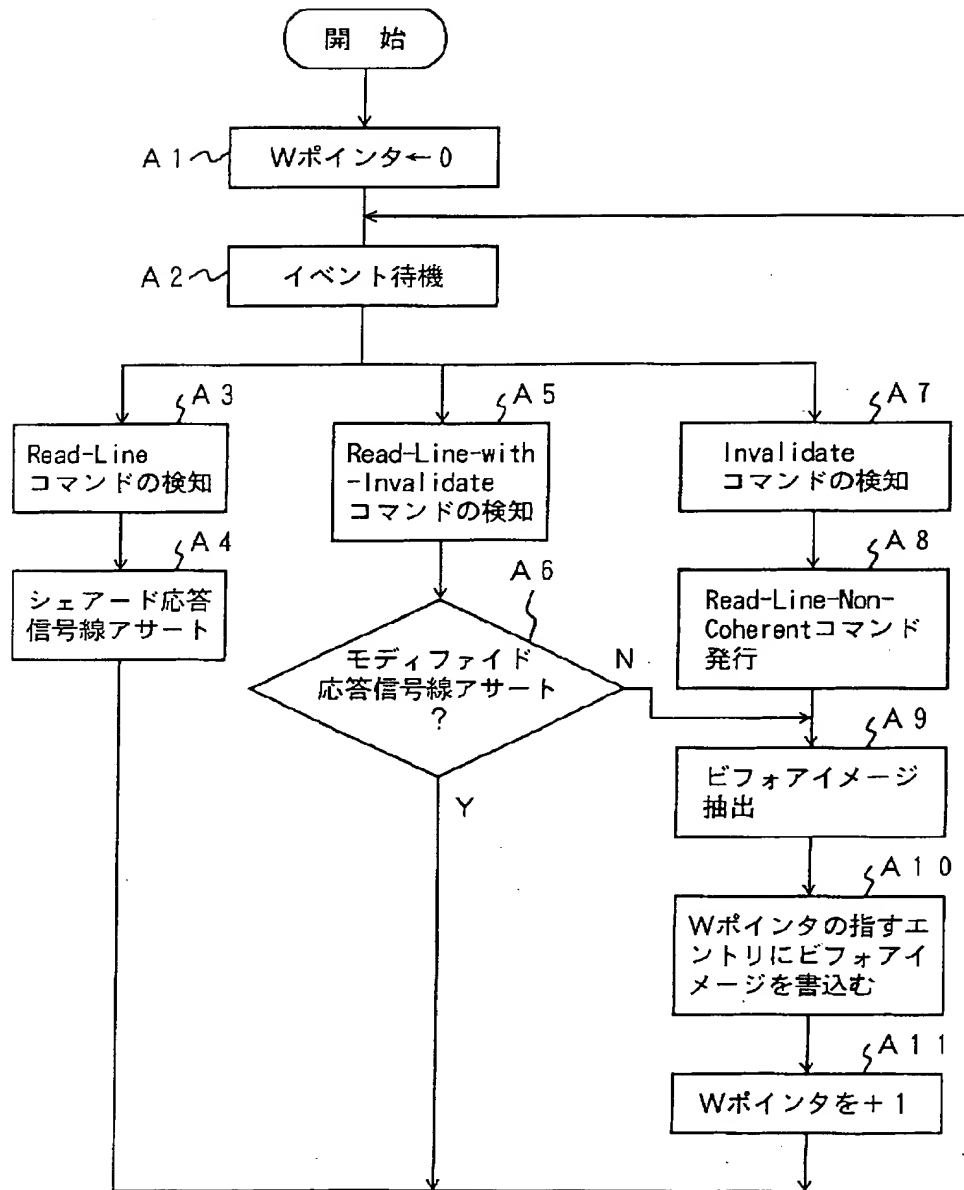
【図 12】



【図 7】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	ビフォア イメージ
1		Clean-Shared a : A 0		a : A 0 b : B 0	無し
2	a 番地更新	Clean-Shared a : A 1	Invalidate (a)	a : A 0 b : B 0	無し
3		Modified a : A 1	応答 : ACK	a : A 0 b : B 0	無し
4		Modified a : A 1	Read-Line-None- Coherent (a)	a : A 0 b : B 0	無し
5		Modified a : A 1	応答 : A 0 fromメインメモリ	a : A 0 b : B 0	a : A 0
6	b 番地更新	Modified a : A 1	Read-Line-with- Invalidate (b)	a : A 0 b : B 0	a : A 0
7		Modified b : B 1	応答 : B 0 fromメインメモリ	a : A 0 b : B 0	a : A 0 b : B 0
8		Modified b : B 1	Write-Line (a : A 1)	a : A 0 b : B 0	a : A 0 b : B 0
9		Modified b : B 1	応答 : ACK	a : A 1 b : B 0	a : A 0 b : B 0
10	a 番地更新	Modified b : B 1	Read-Line-with- Invalidate (a)	a : A 1 b : B 0	a : A 0 b : B 0
11		Modified a : A 2	応答 : A 1 fromメインメモリ	a : A 1 b : B 0	a : A 0 b : B 0 a : A 1
12		Modified a : A 2	Write-Line (b : B 1)	a : A 1 b : B 0	a : A 0 b : B 0 a : A 1
13		Modified a : A 2	応答 : ACK	a : A 1 b : B 1	a : A 0 b : B 0 a : A 1

【図3】



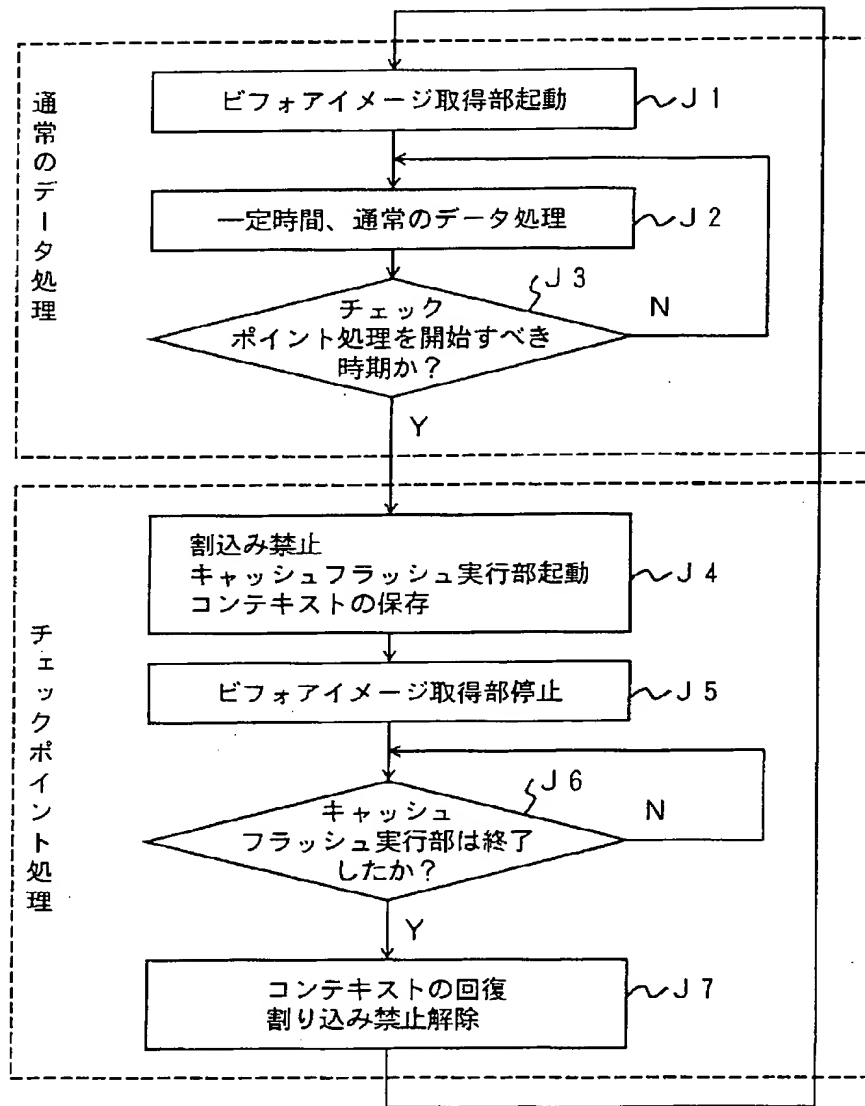
【図14】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	フラグ メモリ	ビフォ イメージ
1		Modified a: A2		a: A1 b: B0	b: オン	a: A0 b: B0
2		Modified a: A2	Read-Line (a)	a: A1 b: B1	b: オフ	a: A0 b: B0
3		Clean-Shared a: A2	応答: A2 from キャッシュ メモリ	a: A2 b: B1	b: オフ	a: A0 b: B0
4		Clean-Shared a: A2	Read-Line (b)	a: A2 b: B1	b: オフ	a: A0 b: B0
5		Clean-Shared a: A2	応答: B1 from メインメモリ	a: A2 b: B1	b: オフ	a: A0 b: B0

【図19】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	カウンタ	ビフォ イメージ
1		Modified a: A2		a: A1 b: B0	1	a: A0 b: B0 a: A1
2		Modified a: A2	Read-Line (a)	a: A1 b: B1	1	a: A0 b: B0 a: A1
3		Clean-Shared a: A2	応答: A2 from キャッシュ メモリ	a: A2 b: B1	0	a: A0 b: B0 a: A1
4		Clean-Shared a: A2		a: A2 b: B1	0	a: A0 b: B0 a: A1
5		Clean-Shared a: A2		a: A2 b: B1	0	a: A0 b: B0 a: A1

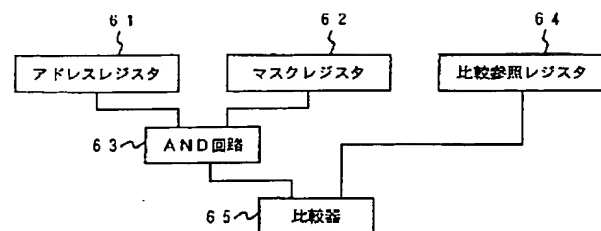
【図6】



【図15】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	フラグ メモリ	ビフォア イメージ
1		Modified a:A2		b:A1 b:B0	a:オン b:オン	b:A0 b:B0
2	プロセッサがキャッシュブロックの無効化命令を実行					
3		Invalid		b:A1 b:B0	a:オン b:オン	b:A0 b:B0
4		Invalid	Write-Line (b:A0)	b:A1 b:B0	a:オン b:オン	a:A0
5		Invalid	応答: ACK	a:A0 b:B0	a:オン b:オン	a:A0
6		Invalid	Write-Line (b:A0)	a:A1 b:B0	a:オン b:オン	なし
7		Invalid	応答: ACK	a:A0 b:B0	a:オン b:オン	なし
8	プロセッサがフラグメモリをすべてオフにする					
9		Invalid		b:A0 b:B0	b:オフ	なし

【図21】



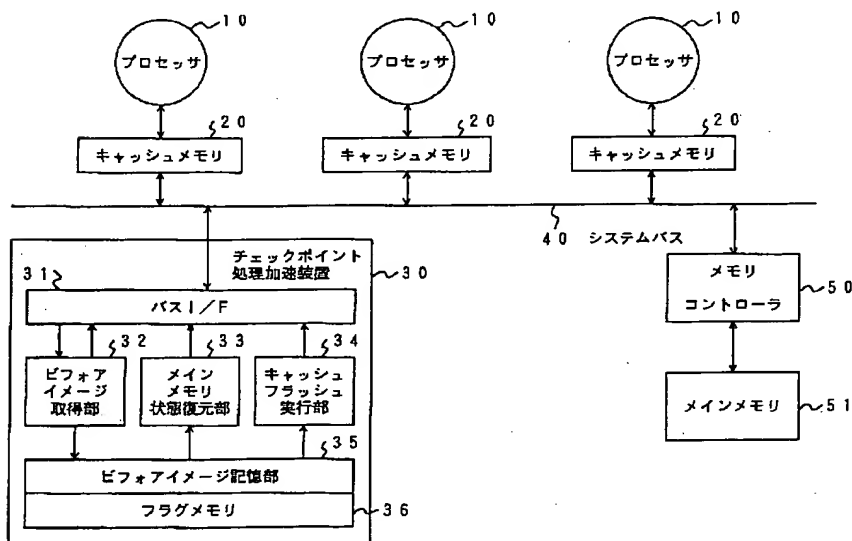
【図8】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	ビフォア イメージ
1		Modified a : A 2		a : A 1 b : B 1	a : A 0 b : B 0 a : A 1
2		Modified a : A 2	Read-Line (a)	a : A 1 b : B 1	a : A 0 b : B 0 a : A 1
3		Clean-Shared a : A 2	応答 : A 2 fromキャッシュ メモリ	a : A 2 b : B 1	a : A 0 b : B 0 a : A 1
4		Clean-Shared a : A 2	Read-Line (b)	a : A 2 b : B 1	a : A 0 b : B 0 a : A 1
5		Clean-Shared a : A 2	応答 : B 1 fromメインメモリ	a : A 2 b : B 1	a : A 0 b : B 0 a : A 1
6		Clean-Shared a : A 2	Read-Line (a)	a : A 2 b : B 1	a : A 0 b : B 0 a : A 1
7		Clean-Shared a : A 2	応答 : A 2 fromメインメモリ	a : A 2 b : B 1	a : A 0 b : B 0 a : A 1

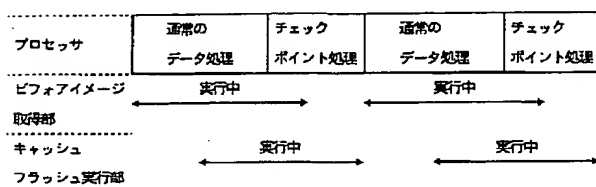
【図9】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	ビフォア イメージ
1		Modified a : A 2		a : A 1 b : B 1	a : A 0 b : B 0 a : A 1
2	(プロセッサがキャッシュブロックの無効化命令を実行)				
3		Invalid		a : A 1 b : B 1	a : A 0 b : B 0 a : A 1
4		Invalid	Write-Line (a : A 1)	a : A 1 b : B 1	a : A 0 b : B 0
5		Invalid	応答 : ACK	a : A 1 b : B 1	a : A 0 b : B 0
6		Invalid	Write-Line (b : B 0)	a : A 1 b : B 1	a : A 0
7		Invalid	応答 : ACK	a : A 1 b : B 0	a : A 0
8		Invalid	Write-Line (a : A 0)	a : A 1 b : B 0	なし
9		Invalid	応答 : ACK	a : A 0 b : B 0	なし

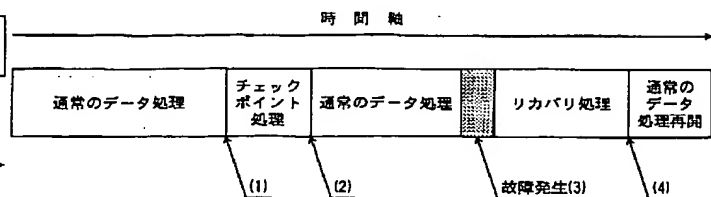
【図10】



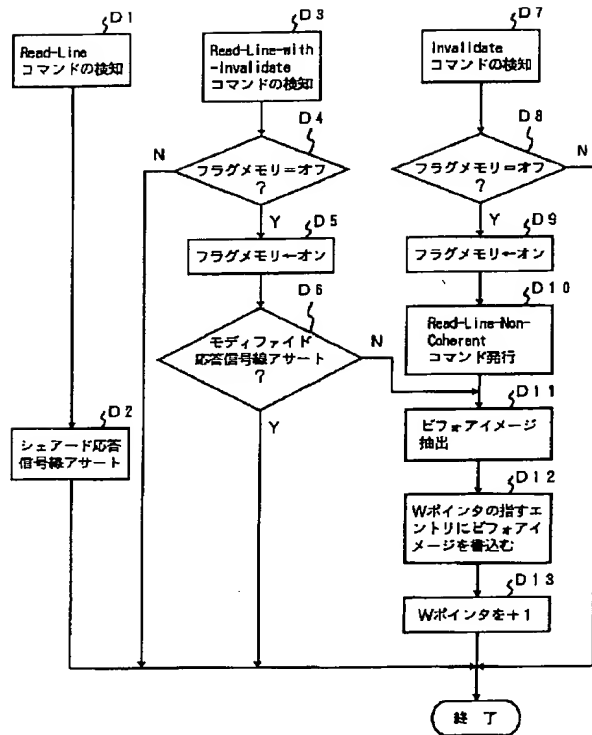
【図22】



【図29】



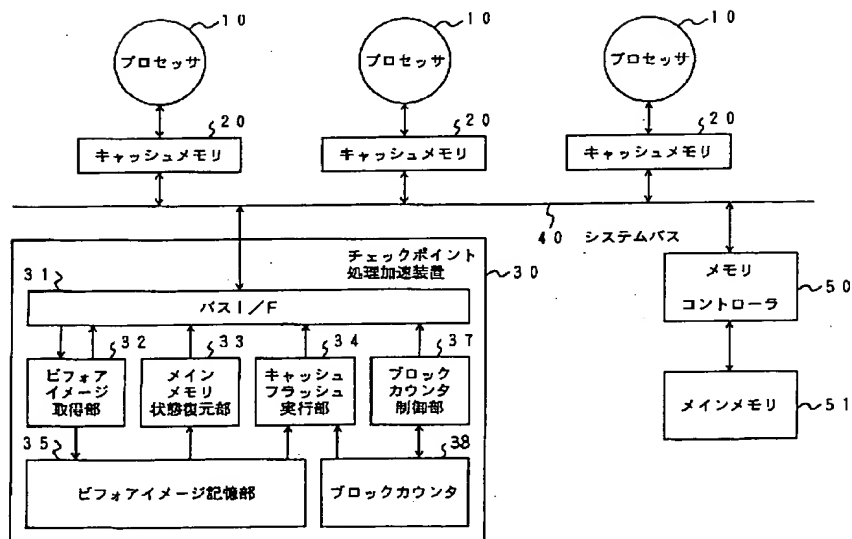
【図11】



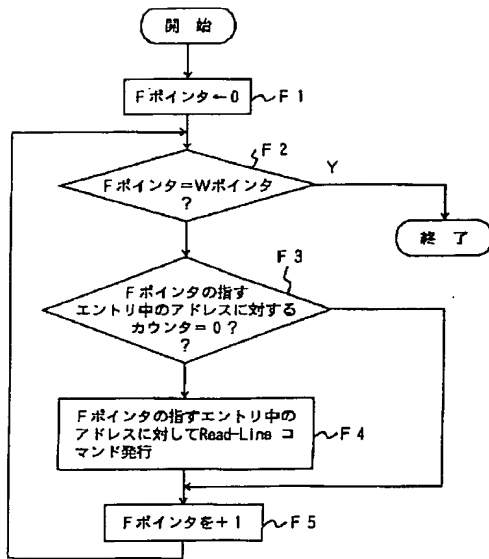
【図13】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	フラグ メモリ	ビフォア イメージ
1		Clean-Shared a : A 0		a : A 0 b : B 0	a : オフ b : オフ	無し
2	a 番地更新	Clean-Shared a : A 0	Invalidate (a)	a : A 0 b : B 0	a : オフ b : オフ	無し
3		Modified a : A 1	応答 : ACK	a : A 0 b : B 0	a : オン b : オフ	無し
4			Read-Line-Non- Coherent (a)	a : A 0 b : B 0	a : オン b : オフ	無し
5			応答 : A 0 fromメインメモリ	a : A 0 b : B 0	a : オン b : オフ	a : A 0
6	b 番地更新	Modified a : A 1	Read-Line-with- Invalidate (b)	a : A 0 b : B 0	a : オン b : オフ	a : A 0
7		Modified b : B 1	応答 : B 0 fromメインメモリ	a : A 0 b : B 0	a : オン b : オン	a : A 0 b : B 0
8		Modified b : B 1	Write-Line (a : A 1)	a : A 0 b : B 0	a : オン b : オン	a : A 0 b : B 0
9		Modified b : B 1	応答 : ACK	a : A 1 b : B 0	a : オン b : オン	a : A 0 b : B 0
10	a 番地更新	Modified b : B 1	Read-Line-with- Invalidate (a)	a : A 1 b : B 0	a : オン b : オン	a : A 0 b : B 0
11		Modified a : A 2	応答 : A 1 fromメインメモリ	a : A 1 b : B 0	a : オン b : オン	a : A 0 b : B 0
12		Modified a : A 2	Write-Line (b : B 1)	a : A 1 b : B 0	a : オン b : オン	a : A 0 b : B 0
13		Modified a : A 2	応答 : ACK	a : A 1 b : B 1	a : オン b : オン	a : A 0 b : B 0

【図16】



【図 17】



【図 18】

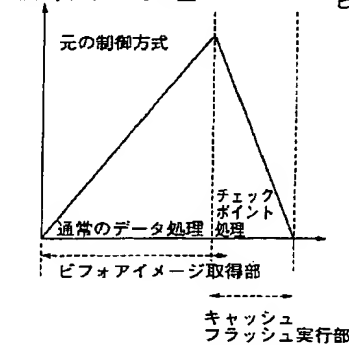
	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	カウンタ	ビフォア イメージ
1		Clean-Shared a : A 0		a : A 0 b : B 0	0	無し
2	a 番地更新	Clean-Shared a : A 0	Invalidate (a)	a : A 0 b : B 0	0	無し
3		Modified a : A 1	応答 : ACK	a : A 0 b : B 0	1	無し
4		Modified a : A 1	Read-Line-None- Coherent (a)	a : A 0 b : B 0	1	無し
5		Modified a : A 1	応答 : A 0 fromメインメモリ	a : A 0 b : B 0	1	a : A 0
6	b 番地更新	Modified a : A 1	Read-Line-with- Invalidate (b)	a : A 0 b : B 0	1	a : A 0
7		Modified b : B 1	応答 : B 0 fromメインメモリ	a : A 0 b : B 0	2	a : A 0 b : B 0
8		Modified b : B 1	Write-Line (a : A 1)	a : A 0 b : B 0	2	a : A 0 b : B 0
9		Modified b : B 1	応答 : ACK	a : A 1 b : B 0	1	a : A 0 b : B 0
10	a 番地更新	Modified b : B 1	Read-Line-with- Invalidate (a)	a : A 1 b : B 0	1	a : A 0 b : B 0
11		Modified a : A 2	応答 : A 1 fromメインメモリ	a : A 1 b : B 0	2	a : A 0 b : B 0 a : B 1
12		Modified a : A 2	Write-Line (b : B 1)	a : A 1 b : B 0	2	a : A 0 b : B 0 a : B 1
13		Modified a : A 2	応答 : ACK	a : A 1 b : B 1	1	a : A 0 b : B 0 a : B 1

【図 20】

【図 24】

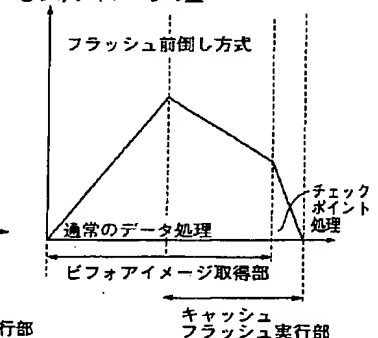
	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	カウンタ	ビフォア イメージ
1		Modified a : A 2		a : A 1 b : B 1	1	a : A 0 b : B 0 a : A 1
2	(プロセッサがキャッシュブロックの無効化命令を実行)					
3		Invalid		a : A 1 b : B 1	1	a : A 0 b : B 0 a : A 1
4		Invalid	Write-Line (a : A 1)	a : A 1 b : B 1	1	a : A 0 b : B 0 a : A 1
5		Invalid	応答 : ACK	a : A 1 b : B 1	1	a : A 0 b : B 0 a : A 1
6		Invalid	Write-Line (b : B 0)	a : A 1 b : B 1	1	a : A 0
7		Invalid	応答 : ACK	a : A 1 b : B 0	1	a : A 0
8		Invalid	Write-Line (a : A 0)	a : A 1 b : B 0	1	なし
9		Invalid	応答 : ACK	a : A 0 b : B 0	1	なし
10	(プロセッサがすべてのカウンタを0にする)					
11		Invalid		a : A 0 b : B 0	0	なし

キャッシュフラッシュ
実行部処理すべき
ビフォアイメージの量



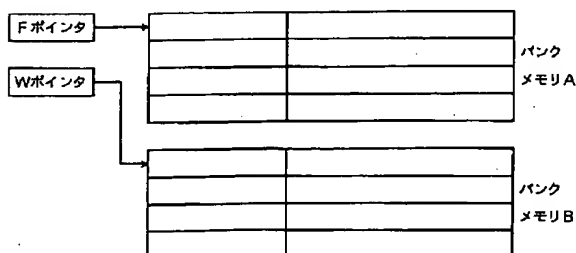
(a)

キャッシュフラッシュ
実行部処理すべき
ビフォアイメージの量



(b)

【図 26】




```

graph TD
    Start(( )) --> G1[G1: ビフォーアイメージ取得部起動]
    G1 --> G2[G2: 一定時間、通常のデータ処理]
    G2 --> G3{G3: キャッシュフラッシュ実行部を起動すべき時期か?}
    G3 -- N --> G2
    G3 -- Y --> G4[G4: キャッシュフラッシュ実行部起動]
    G4 --> G5[G5: 一定時間、通常のデータ処理]
    G5 --> G6{G6: チェックポイント処理を開始すべき時期か?}
    G6 -- N --> G5
    G6 -- Y --> G7[G7: 割り込み禁止コンテキストの保存]
    G7 --> G8[G8: ビフォーアイメージ取得部停止]
    G8 --> G9{G9: キャッシュフラッシュ実行部は終了したか?}
    G9 -- N --> G8
    G9 -- Y --> G10[G10: コンテキストの回復  
割り込み禁止解除]
    G10 --> End(( ))
  
```

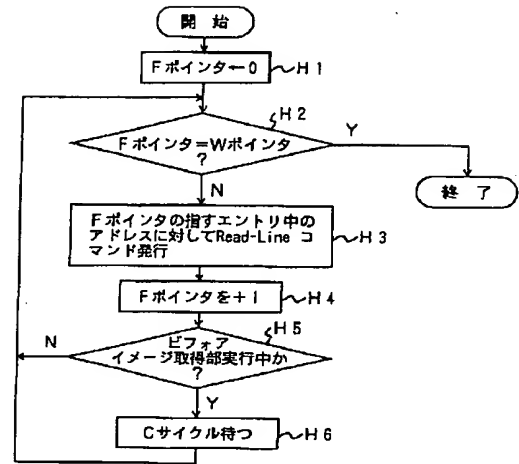
通常のデータ処理

チェックポイント処理

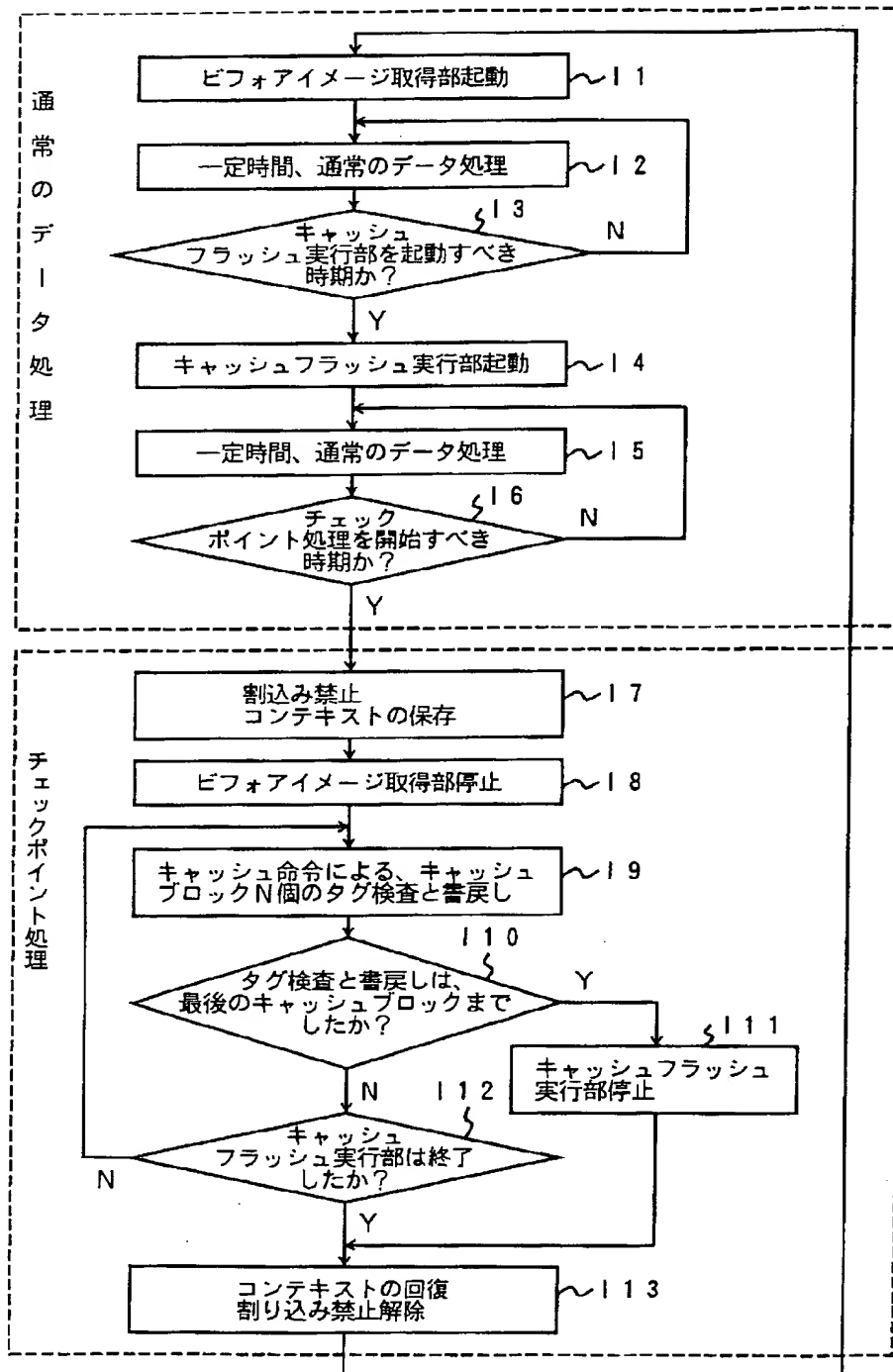
【図25】

	プロセッサ	キャッシュ ブロックBLK0	システムバス	メイン メモリ	フラグ メモリ	ビフォア イメージ
1		Clean-Shared a : A 0		a : A 0 b : B 0	b : オフ	無し
2	a 番地更新	Clean-Shared a : A 0	Invalidate (a)	a : A 0 b : B 0	b : オフ	無し
3		Modified a : A 1	応答 : ACK	a : A 0 b : B 0	b : オフ	無し
4		Modified a : A 1	Read-Line-None- Coherent (a)	a : A 0 b : B 0	b : オフ	無し
5		Modified a : A 1	応答 : A 0 fromメインメモリ	a : A 0 b : B 0	b : オン	a : A 0
6	b 番地更新	Modified a : A 1	Read-Line-with- Invalidate (b)	a : A 0 b : B 0	b : オン	a : A 0
7		Modified b : B 1	応答 : B 0 fromメインメモリ	a : A 0 b : B 0	b : オン	a : A 0
8		Modified b : B 1	Write-Line (a : A 1)	a : A 0 b : B 0	b : オン	a : A 0
9		Modified b : B 1	応答 : ACK	a : A 1 b : B 0	b : オン	a : A 0
10	(プロセッサがキャッシュフラッシュ実行部を起動)					
11		Modified b : B 1	Read-Line (a)	a : A 1 b : B 0	b : オン	a : A 0 b : B 0
12		Modified b : B 1	応答 : A 1 fromメインメモリ	a : A 1 b : B 0	b : オン	a : A 0 b : B 0
13		Modified b : B 1	Read-Line (b)	a : A 1 b : B 0	b : オン	a : A 0 b : B 0
14		Clean-Shared b : B 1	応答 : B 1 fromキャッシュ メモリ	a : A 1 b : B 1	b : オン	a : A 0 b : B 0
15	a 番地更新	Clean-Shared b : B 1	Read-Line-with- Invalidate (a)	a : A 1 b : B 1	b : オン	a : A 0 b : B 0
16		Modified a : A 2	応答 : A 1 fromメインメモリ	a : A 1 b : B 1	b : オン	a : A 0 b : B 0 a : A 1
17		Modified a : A 2	Read-Line (a)	a : A 1 b : B 1	b : オン	a : A 0 b : B 1 a : A 1
18		Clean-Shared a : A 2	応答 : A 2 fromキャッシュ メモリ	a : A 2 b : B 1	b : オン	a : A 0 b : B 1 a : A 1

【図27】



【図28】



【公報種別】特許法第 17 条の 2 の規定による補正の掲載
 【部門区分】第 6 部門第 3 区分
 【発行日】平成 11 年（1999）11 月 5 日

【公開番号】特開平 10-78918
 【公開日】平成 10 年（1998）3 月 24 日
 【年通号数】公開特許公報 10-790
 【出願番号】特願平 8-234321
 【国際特許分類第 6 版】

G06F 12/08 310
 11/14 310
 12/16 310

【F I】

G06F 12/08 310 B
 11/14 310 B
 12/16 310 C

【手続補正書】

【提出日】平成 11 年 1 月 20 日

【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

【補正内容】

【発明の名称】 チェックポイント処理加速装

置およびデータ処理方法

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項 1】 バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、前記プロセッサと前記メインメモリとを接続するシステムバスとを具備してなる計算機に適用されるチェックポイント処理加速装置であって、

データ更新が発生したアドレスおよびその更新前のデータを組としたビフォアイメージを複数個記憶するビフォアイメージ記憶手段と、

前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答を前記システムバスの監視によって検知し、前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答が更新アドレスおよび更新前データを含む場合に、そのアドレスおよび更新前データを前記ビフォアイメージ記憶手段に格納し、前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答が更新の発生したアドレスを含むが更新前のデ

ータを含まない場合に、その更新前のデータを読み出すためのコマンドを前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答に含まれるアドレスを用いて前記システムバスに発行し、前記アドレスおよび読み出された更新前のデータを前記ビフォアイメージ記憶手段に格納するビフォアイメージ取得手段と、

前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスで示される更新状態のデータをメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行するキャッシュフラッシュ実行手段と、前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべての更新前データについて、新しく格納したものの順にメインメモリへの更新書き込みを要求するコマンドを前記システムバスに発行するメインメモリ状態復元手段とを具備してなることを特徴とするチェックポイント処理加速装置。

【請求項 2】 前記メインメモリを前記キャッシュメモリ内に確保されるキャッシュブロックの大きさと分割し、この分割によって定義される複数の区画の中の少なくとも一つ以上の区画それぞれに対応させて、オンとオフとの二つの状態が割り当てられるフラグメモリをさらに設け、

前記ビフォアイメージ取得手段は、前記ビフォアイメージを取得した際、その更新アドレスに対するフラグメモリが設けられ、かつオフのときには、前記フラグメモリをオンに設定するとともに前記ビフォアイメージを前記ビフォアイメージ記憶手段に格納し、更新アドレスに対するフラグメモリが設けられ、かつオンのときには、前記ビフォアイメージの前記ビフォアイメージ記憶手段への格納を行わず、更新アドレスに対するフラグメモリ

が設けられていないときには、前記ビフォアイメージをビフォアイメージ記憶手段に格納する手段を具備し、前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段からビフォアイメージを取り出して、更新状態にあるキャッシュブロックの内容の前記メインメモリへの書き戻しを要求するコマンドを発行する際、そのアドレスに対応するフラグメモリが設けられているときには、そのフラグメモリをオフに設定する手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項3】 互いに共通部分をもたないように設定される一つ以上のアドレス範囲それぞれに対応して設けられ、前記アドレス範囲に属する更新状態のキャッシュブロック数を保持するブロックカウンタと、前記システムバス上のコマンドおよびその応答を監視して、前記キャッシュブロックが更新状態に移行することを検知した際、前記システムバス上のコマンドおよびその応答がアクセスするアドレスに対応した前記ブロックカウンタが存在するときは、そのブロックカウンタをインクリメントし、前記キャッシュブロックが更新状態からそれ以外の状態に移行すること検知した際、前記システムバス上のコマンドおよびその応答がアクセスするアドレスに対応した前記ブロックカウンタがあるときは、そのブロックカウンタをデクリメントするカウンタ制御手段とをさらに具備し、

前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスに対応する前記ブロックカウンタがないか、またはその値が初期値でないときに、そのアドレスの更新状態のキャッシュブロックの内容をメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行する手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項4】 前記キャッシュメモリは、ダイレクトマップで構成され、前記ブロックカウンタは、キャッシュブロックと一対に設けられることを特徴とする請求項3記載のチェックポイント処理加速装置。

【請求項5】 前記キャッシュメモリは、 n ウェイのセットアソシアティブで構成され、前記ブロックカウンタは、 n 個のキャッシュブロックで形成されるグループと一対に設けられることを特徴とする請求項3記載のチェックポイント処理加速装置。

【請求項6】 前記メインメモリを前記キャッシュメモリ内に確保されるキャッシュブロックの大きさで分割し、この分割によって定義される複数の区画の中の少なくとも一つ以上の区画それぞれに対応させて、オンとオフとの二つの状態が割り当てられるフラグメモリをさらに設け、前記システムバス上のコマンドおよびその応答を監視して、前記キャッシュブロックが更新状態に移行すること

を検知した際、前記システムバス上のコマンドおよびその応答がアクセスするアドレスに対応した前記フラグメモリが設けられているときは、そのフラグメモリをオンに設定し、前記キャッシュブロックが更新状態からそれ以外の状態に移行すること検知した際、前記システムバス上のコマンドおよびその応答がアクセスするアドレスに対応した前記フラグメモリが設けられているときは、そのフラグメモリをオフに設定するフラグメモリ制御手段をさらに具備し、

前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスに対応する前記フラグメモリが設けられていないか、またはその状態がオンのときに、そのアドレスの更新状態のキャッシュブロックの内容をメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行する手段を具備してなることを特徴とする請求項1記載のチェックポイント処理加速装置。

【請求項7】 前記ビフォアイメージ取得手段が前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドまたはそれに対する応答を検知したときに、そのビフォアイメージを前記ビフォアイメージ記憶手段に格納するか否かをその更新アドレスにより判断するアドレス判定手段をさらに具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項8】 前記ビフォアイメージ記憶手段に格納された前記ビフォアイメージそれぞれについて、前記メインメモリ状態復元手段により前記メインメモリに書き戻すか否かをその更新アドレスにより判断するアドレス判定手段をさらに具備してなることを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項9】 前記キャッシュメモリ上でのデータ更新が行なわれた際、その更新対象のデータが他のキャッシュメモリ上で更新されて保持されているデータであるときに、前記ビフォアイメージ取得手段は、前記ビフォアイメージを前記ビフォアイメージ記憶手段に格納しないことを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項10】 前記キャッシュメモリを構成する各キャッシュブロックに割り当てられる複数の管理状態の中に、そのキャッシュブロックが保持しているデータは他のプロセッサのキャッシュメモリには保持されておらず、かつメインメモリ上と同じデータであるといったクリーンエクスクルーシブ状態を含むものであり、かつ、いずれかのキャッシュメモリの発行したメインメモリデータの読み込み要求コマンドに対し、そのデータを保持していると応答する応答手段をさらに具備することを特徴とする請求項1、2、3、4、5または6記載のチェックポイント処理加速装置。

【請求項 11】 前記ビフォアイメージ取得手段は、前記キャッシュメモリの機構を介さずに前記メインメモリの内容を更新するコマンドを検知したときに、そのコマンドの実行を中止させ、前記キャッシュメモリの機構を介さずに前記メインメモリの内容を更新するコマンドに含まれる更新アドレスを用いて前記メインメモリから更新前のデータを読み出すためのコマンドを前記システムバスに発行し、前記アドレスと読み出された更新前のデータとを前記ビフォアイメージ格納手段に格納する手段と、前記実行を中止させたコマンドが再度発行されたときには、そのコマンドの実行の中止を行なわない手段とを具備してなることを特徴とする請求項 1、2、3、4、5 または 6 記載のチェックポイント処理加速装置。

【請求項 12】 前記ビフォアイメージ取得手段は、前記プロセッサから指示されたアドレス範囲について、前記メインメモリから更新前のデータを読み出すためのコマンドを前記システムバスに発行し、前記アドレスとその読み出された更新前のデータを組としたビフォアイメージを前記ビフォアイメージ格納手段に格納する手段を具備してなることを特徴とする請求項 1、2、3、4、5 または 6 記載のチェックポイント処理加速装置。

【請求項 13】 前記ビフォアイメージ取得手段およびキャッシュフラッシュ実行手段は、並行して実行可能であることを特徴とする請求項 1、2 または 3 記載のチェックポイント処理加速装置。

【請求項 14】 前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ記憶手段に格納されたビフォアイメージの中の最も早い時点で取得されたものから順に処理することを特徴とする請求項 13 記載のチェックポイント処理加速装置。

【請求項 15】 前記ビフォアイメージ記憶手段は、互いに独立した複数のメモリで構成されることを特徴とする請求項 13 記載のチェックポイント処理加速装置。

【請求項 16】 前記ビフォアイメージ記憶手段は、互いに独立した第 1 および第 2 のメモリを含んで構成され、前記ビフォアイメージ取得手段は、前記キャッシュフラッシュ実行手段が動作していないときには、前記取得したビフォアイメージを前記第 1 のメモリに格納し、前記キャッシュフラッシュ実行手段が動作を開始したときには、前記取得したビフォアイメージを前記第 2 のメモリに格納する手段を具備し、前記キャッシュフラッシュ実行手段は、前記第 1 のメモリに格納されたビフォアイメージから処理する手段を具備してなることを特徴とする請求項 13 記載のチェックポイント処理加速装置。

【請求項 17】 前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ取得手段によって前記第 1 のメモリのすべてに前記ビフォアイメージが格納されたときに起動されることを特徴とする請求項 16 記載のチェ

ックポイント処理加速装置。

【請求項 18】 前記ビフォアイメージ記憶手段の残容量が予め定められた量を下回ったことを前記プロセッサに通知する通知手段をさらに具備してなることを特徴とする請求項 1、2、3、4、5 または 6 記載のチェックポイント処理加速装置。

【請求項 19】 前記キャッシュフラッシュ実行手段は、前記ビフォアイメージ取得手段が実行中であるときに、更新状態のキャッシュブロックの内容を前記メインメモリへ書き戻すことを要求するコマンドの発行頻度を予め定められた値以内に抑制し、前記ビフォアイメージ取得手段が停止したときに、前記コマンドの発行頻度を高くする手段を具備してなることを特徴とする請求項 13 記載のチェックポイント処理加速装置。

【請求項 20】 バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、請求項 1 乃至 19 のいずれかに記載のチェックポイント処理加速装置と、前記プロセッサと前記メインメモリと前記チェックポイント処理加速装置とを接続するシステムバスを備える計算機システムにおけるデータ処理方法であって、

前記チェックポイント処理加速装置のビフォアイメージ取得手段を起動した後、通常のデータ処理を実行するデータ処理実行ステップと、

前記データ処理実行ステップを一時中断して、すべての前記プロセッサが同期して実行する、データ処理中のコンテキストをメインメモリに格納して、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段を起動し、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段の実行終了後、前記データ処理実行ステップを再開するチェックポイント作成ステップと、

故障が発生したときに、すべての前記プロセッサが同期して前記キャッシュメモリのすべてのキャッシュブロックを無効化した後、前記チェックポイント処理加速装置のメインメモリ状態復元手段を起動することにより前記メインメモリの状態を直前に採取したチェックポイント時点の状態に復元し、その後前記データ処理実行ステップを再開するロールバック／リカバリステップとを有することを特徴とするデータ処理方法。

【請求項 21】 バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、請求項 1 乃至 19 のいずれかに記載のチェックポイント処理加速装置と、前記プロセッサと前記メインメモリと前記チェックポイント処理加速装置とを接続するシステムバスとを備える計算機システムにおけるデータ処理方法であって、前記チェックポイント処理加速装置のビフォアイメージ取得手段を起動した後、通常のデータ処理を実行すると共に、前記チェックポイント処理加速装置のキャッシュ

フラッシュ実行手段の起動も行うデータ処理実行ステップと、

前記データ処理実行ステップを一時中断して、すべての前記プロセッサが同期して実行する、データ処理中のコンテキストをメインメモリに格納して、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段の実行終了後、前記データ処理実行ステップを再開するチェックポイント作成ステップと、

故障が発生したときに、すべての前記プロセッサが同期して前記キャッシュメモリのすべてのキャッシュブロックを無効化した後、前記チェックポイント処理加速装置のメインメモリ状態復元手段を起動することにより前記メインメモリの状態を直前に採取したチェックポイント時点の状態に復元し、その後前記データ処理実行ステップを再開するロールバック／リカバリステップとを有することを特徴とするデータ処理方法。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正内容】

【0001】

【発明の属する技術分野】この発明は、たとえばデータの一貫性を保証するためにスヌープ機構を有してなるキャッシュメモリを備えた計算機に適用して好適なチェックポイント加速装置およびデータ処理方法に関する。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0024

【補正方法】変更

【補正内容】

【0024】この発明は、このような実情に鑑みてなされたものであり、特別なキャッシュメモリを必要とせず、標準的なユニプロセッサあるいはマルチプロセッサを搭載する計算機でチェックポイント／リカバリ機能を容易に実現可能とするチェックポイント処理加速装置およびデータ処理方法を提供することを目的とする。これにより、標準的な計算機における性能向上の恩恵を享受することができ、また、USP4,740,969に開示される方法と比較して、通常のデータ処理において取得するデータの量がより少なく、チェックポイント処理時のキャッシュフラッシュ機能も実現するチェックポイント処理加速装置およびデータ処理方法を提供することが可能となる。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0025

【補正方法】変更

【補正内容】

【0025】この発明のチェックポイント処理加速装置

は、バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、前記プロセッサと前記メインメモリとを接続するシステムバスとを具備してなる計算機に適用されるチェックポイント処理加速装置であって、データ更新が発生したアドレスおよびその更新前のデータを組としたビフォアイメージを複数個記憶するビフォアイメージ記憶手段と、前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答を前記システムバスの監視によって検知し、前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答が更新アドレスおよび更新前データを含む場合に、そのアドレスおよび更新前データを前記ビフォアイメージ記憶手段に格納し、前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答が更新の発生したアドレスを含むが更新前のデータを含まない場合に、その更新前のデータを読み出すためのコマンドを前記キャッシュメモリ上でデータ更新が発生したことを示すコマンドおよびそれに対する応答に含まれるアドレスを用いて前記システムバスに発行し、前記アドレスおよび読み出された更新前のデータを前記ビフォアイメージ記憶手段に格納するビフォアイメージ取得手段と、前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべてのアドレスについて、そのアドレスで示される更新状態のデータをメインメモリに書き戻すことを要求するコマンドを前記システムバスに発行するキャッシュフラッシュ実行手段と、前記プロセッサから指示されたときに、前記ビフォアイメージ記憶手段に格納されたすべての更新前データについて、新しく格納したもののから順にメインメモリへの更新書き込みを要求するコマンドを前記システムバスに発行するメインメモリ状態復元手段とを具備してなることを特徴とする。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0030

【補正方法】変更

【補正内容】

【0030】また、この発明のデータ処理方法は、バススヌープ機構を有するコピーバック型のキャッシュメモリを備えた少なくとも一つ以上のプロセッサと、メインメモリと、請求項1乃至19のいずれかに記載のチェックポイント処理加速装置と、前記プロセッサと前記メインメモリ前記チェックポイント処理加速装置とを接続するシステムバスを備える計算機システムにおけるデータ処理方法であって、前記チェックポイント処理加速装置のビフォアイメージ取得手段を起動した後、通常のデータ処理を実行するデータ処理実行ステップと、前記データ処理実行ステップを一時中断して、すべての前記プロセッサが同期して実行する、データ処理中のコンテキス

トをメインメモリに格納して、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段を起動し、前記チェックポイント処理加速装置のキャッシュフラッシュ実行手段の実行終了後、前記データ処理実行ステップを再開するチェックポイント作成ステップと、故障が発生したときに、すべての前記プロセッサが同期して前記キャッシュメモリのすべてのキャッシュブロックを無

効化した後、前記チェックポイント処理加速装置のメインメモリ状態復元手段を起動することにより前記メインメモリの状態を直前に採取してチェックポイント時点の状態に復元し、その後前記データ処理実行ステップを再開するロールバック／リカバリステップとを有すること
を特徴とする。